Part A (Graphical Awareness)

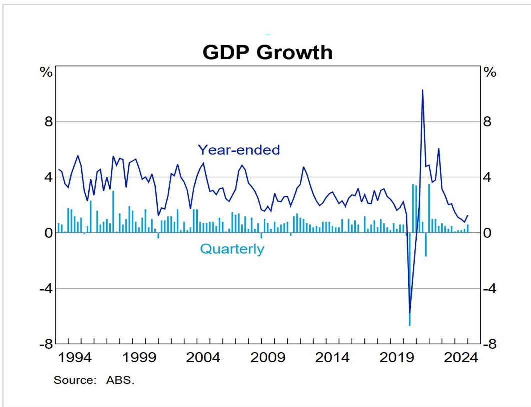
Graphic 1:

Graphic Title: GDP Growth – Year-ended and Quarterly

Organization: Reserve Bank of Australia (RBA)

Page: Australian Economic Growth (Chart Pack)

Link:https://www.rba.gov.au/chart-pack/au-growth.html



Purpose:

The graph shows both quarterly and year-ended GDP growth in Australia from 1994 to 2024, it allows viewers to assess both short-term fluctuations (quarterly) and longer-term trends (year-ended).

Strengths:

1. Well-proportioned Y-axis allows both sharp and subtle trends to be observed:

The Y-axis has a suitable range that not only includes extreme values like those during the COVID-19 period, but also makes small changes easy to see. This helps the viewer observe both sharp shocks and more gradual shifts in year-ended GDP growth over time.

2. Effective use of a line graph to show trends

Using a line graph (instead of bar chart or scatterplot) makes it easier to follow the changes in GDP over time. The continuous line helps highlight not only large movements, but also subtle long-term trends and turning points. This format works especially well for tracking economic data across multiple decades.

Disadvantages:

1. Colour overlap makes it hard to read:

The two series use similar shades of blue. The lighter bars for quarterly data are hard to see when they appear behind the darker line for year-ended growth.

2. Quarterly data is hard to interpret:

Because the Y-axis is shared and suited more for the year-ended series, the small movements in the quarterly data are hard to notice. This makes the short-term pattern less visible.

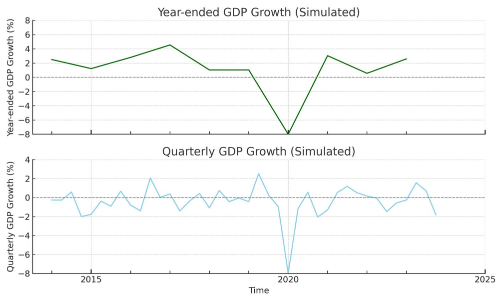
3. Two series may not need to be in one graph:

The quarterly and year-ended series are based on different time frames, so their values are not directly comparable. Quarterly growth reflects short-term changes, while year-ended growth represents longer-term trends. Combining them in one graph may confuse viewers, as the two measures are calculated over different time intervals.

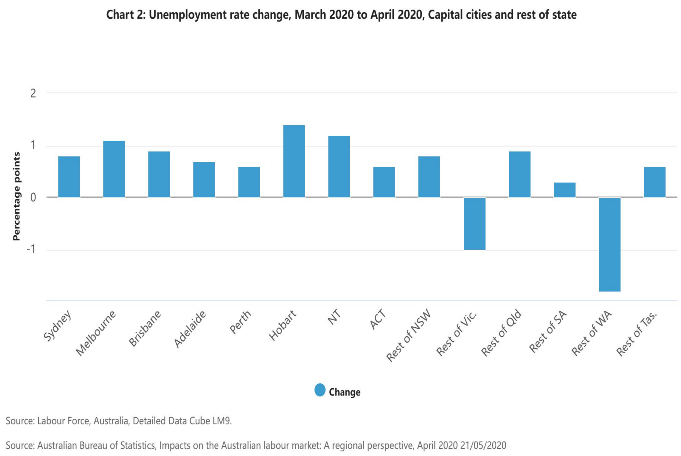
Improved version：

I separate the two series into two plots to make them easier to read. Each plot has its own Y-axis, a clear zero line, and uses different colours. Both are shown as line graphs with the same time axis.The Y values are not directly comparable, but placing the two plots one above the other makes it easy to see short-term and long-term movements over the same time.

Below is a mock-up demonstrating how the graph could be improved,using simulated values.



Graphic 2:



Link:https://www.abs.gov.au/articles/impacts-australian-labour-market-regional-perspective-april-2020

Purpose:
The graph shows the unemployment rate change from March 2020 to April 2020, imcluding data for capital cities and rest of state.The purpose is to highlight the short-term impact of the early COVID-19 shock on the labour market in different regions, using vertical bars to indicate both positive and negative shifts.

The strengths of the graphic:
1. The bar chart format makes it easy to compare unemployment rate changes across regions:
The use of vertical bars allows viewers to quickly identify which areas experienced increases or decreases in unemployment. The visual separation between positive and negative values is also helpful.

The weakness of the graphic:
1. Small differences are hard to compare due to lack of detail:

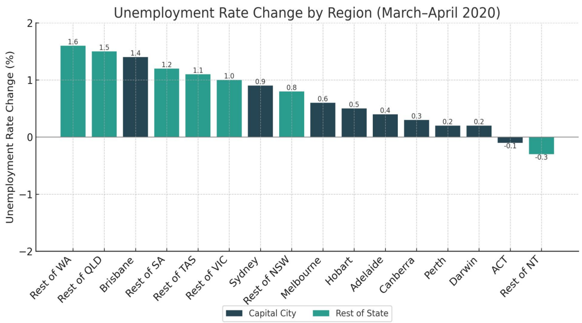
Most values are close together (between -2 and 2), but the Y-axis has only a few tick marks and there are no value labels on the bars. This makes it difficult to see which region had a slightly higher or lower change, especially for similar cases like Perth and ACT.

2. Lack of value-based ordering reduces comparability:
The current order seems arbitrary, making it harder to identify which regions had the largest or smallest changes. Sorting the bars from highest to lowest would improve readability.

3. No visual grouping between capital and regional areas:
Capital cities and rest-of-state areas are mixed together without clear grouping or colour distinction. This makes the chart harder to read and compare, especially when trying to understand differences between urban and regional areas.

Improved version：
To make the graph clearer,the bars should be sorted from highest to lowest to highlight which regions experienced the most change. I would also add value labels on each bar to make small differences easier to see, since most values fall within a narrow range. In addition, using colour grouping to separate capital cities from rest-of-state areas would improve readability and interpretation.

The chart below is a mock-up to demonstrate the proposed improvements. The values are simulated and do not reflect the actual unemployment rate changes from the original graph.

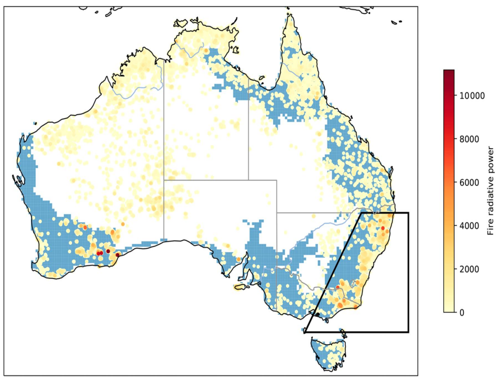


Graphic 3

GraphicTitle:Figure 1Moderate Resolution Imaging Spectroradiometer (MODIS) active fire data

Source:Attribution of the Australian bushfire risk to anthropogenic climate change

Link:https://nhess.copernicus.org/articles/21/941/2021/



Purpose:

This map shows the severity and distribution of bushfires in Australia from 1 October 2019 to 10 January 2020, based on satellite data from the MODIS sensor (Collection 6). The red and yellow points represent active fire detections, with darker colours indicating more severe fires. The blue grid shows forested areas, and the black polygon highlights the specific region analysed in the article. The purpose of the graphic is to provide spatial context for the study, which focuses on fire activity in southeastern Australia's forested areas. Non-forested regions, such as grasslands, are excluded from the analysis due to their different fire behaviour.

Strengths:
1.Map format shows fire locations clearly, supporting interpretation:
Using a map makes it easy to understand where fires occurred across Australia. It helps highlight which regions were most affected.

2.Colour gradient effectively encodes fire severity:
The yellow-to-red colours help viewers quickly compare how strong each fire was. Darker red points represent more severe fires.

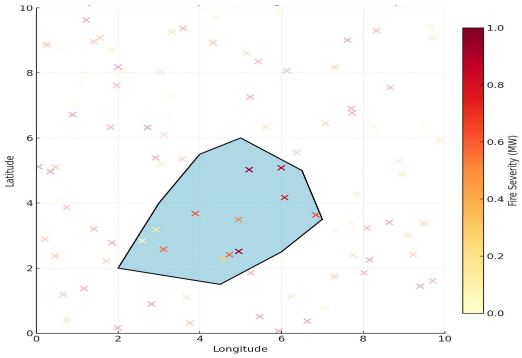
Weakness:
1. Overlapping low-severity fire points make it hard to assess frequency:
In some areas, many light-yellow points representing low-severity fires appear clustered together. Because they are light in colour and often overlap with one another, it becomes difficult to count or estimate how many fire events occurred in those locations. This limits the viewer's ability to fully understand the frequency and distribution of less severe fires.

2.No units on the severity scale:
The colour bar indicates increasing fire severity, but lacks any unit or explanation (e.g. radiative power in MW). This weakens interpretability and conflicts with the idea that graphics must be grounded in good statistics and clear measurement.

3. No clear distinction between fires inside and outside forested areas:
The map does not clearly show which fire points occurred within forested zones. Many fire points overlap the blue forest background, making it hard to see where forest boundaries are. As a result, it becomes difficult to tell whether a fire was inside a forest or not, even though the study focuses only on forested areas. In addition, fire points outside the forest draw attention away from the main area of interest and reduce the clarity of the map's message.

Improved version:
To improve the graphic, each fire point could be shown as an "x"-shaped marker with a dark outline to make overlapping fires easier to see, especially in areas with many low-severity events. The severity colour bar should also include a unit, such as "MW", to clarify what the colours represent. In addition, a visible outline could be added around the forested area to help viewers judge whether fires occurred inside or outside it. Fire points outside the forest zone could be shown with reduced opacity to focus attention on the main study region.

Below is the mocked improved version of the original graphic, based on the suggestions discussed above.

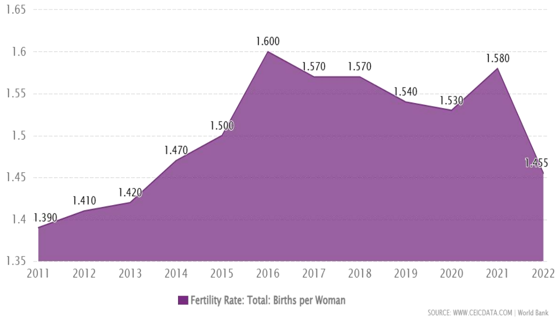


Graphic 4

Link:https://www.ceicdata.com/en/germany/social-health-statistics/de-fertility-rate-total-births-per-woman

Article:Germany DE: Fertility Rate: Total: Births per Woman

Graphic Title:Births per Woman from 2011 to 2022



Purpose:

This graph displays Germany's total fertility rate from 2011 to 2022, indicating the average number of children a woman is expected to have in her lifetime based on age-specific fertility rates in a given year.

Strengths:

1.Enhanced visibility of subtle trends:

The Y-axis spans a narrow range (1.35 to 1.65), which effectively magnifies small year-to-year changes and makes short-term fluctuations in the fertility rate easy to observe. This helps highlight nuanced trends that might otherwise be overlooked.

2.Direct value labels:

Each data point is labelled with its exact value, eliminating the need to estimate from the axis and making the chart highly readable.

3.Clear visual structure:

The area-fill combined with a continuous time series helps the viewer quickly identify the overall pattern of change.

Weaknesses:

1.Potential distortion due to non-zero Y-axis:

The Y-axis does not start at zero, which may exaggerate the perceived increase in fertility rate.

This violates the principle that graphs should represent quantities honestly. For instance, the rise from 1.39 (2011) to 1.47 (2014) appears visually steep despite being only a ~6% increase. The filled area further amplifies this perception due to area-based exaggeration, reducing graphical integrity.

2.Absence of a relevant benchmark:
The graph lacks a contextual reference line such as the replacement fertility rate (commonly 2.1 births per woman), making it hard for viewers to assess whether the values are above or below a meaningful threshold. This undermines interpretability and makes it harder to draw informed conclusions from the chart.

3.No self-contained chart title:
The graph does not include an informative title, which weakens its standalone clarity. According to best practices, a good visualisation should convey its key message without requiring additional explanation, especially for audiences who are skimming or viewing the graph out of context.

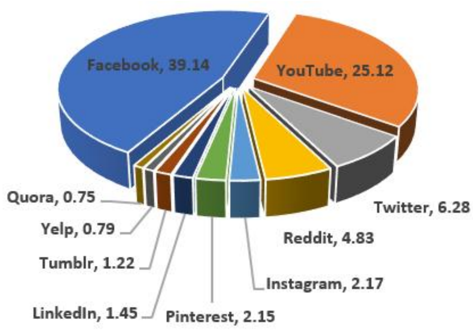
Improved version:To improve the original visualisation, several changes were made. The Y-axis has been expanded to start from 0.8 instead of 1.35 to reduce visual exaggeration and provide a more neutral frame of reference. While the purple area shading is retained to highlight the trend over time, the broader Y-axis range ensures that the fill does not overstate changes. A clear chart title has been added to make the graph self-explanatory without relying on external context. All yearly values are directly labelled above their respective points, offering both trend recognition and precise numerical interpretation. Additionally, a reference line at 2.1 births per woman was added to indicate the replacement fertility level, providing essential context for interpreting whether the fertility rate is above

Graphic 5

Link:https://scc.ms.unimelb.edu.au/resources/data-visualisation-and-exploration /no_pie-charts

Graphic Title:Market share of visits to social network sites in November, 2017

Article:Why you shouldn’t use pie charts



The graphs are reproduced from the University of Queensland's statistical visualisation material. I use them here to demonstrate the common issues with pie charts and how a bar chart can present the same information more clearly. The weaknesses and improvements discussed below are based on my understanding of principles taught in STAT3011.

Purpose:
The bar chart shows the market share of visits to social network sites in November, 2017.

Strengths:
1. Visually engaging and accessible
The colourful design attracts attention and makes the chart feel more approachable and less technical, which can help engage a general audience.

2. Easy to distinguish categories
Each slice is uniquely coloured and directly labelled with percentages, making it easy to identify different social media platforms.

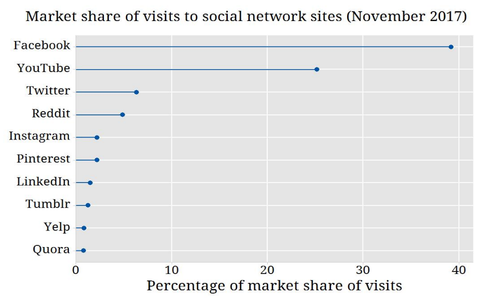
Weaknesses:
1.Difficult to compare proportions
The use of angles makes it hard to accurately compare values between slices. This violates the principle that graphics should facilitate and encourage comparison.

2.Lack of sorting
Slices are not arranged in any meaningful order (e.g. descending), which reduces clarity and makes it harder to interpret the relative sizes of categories.

3.Cluttered with too many categories
The presence of many small segments leads to visual clutter. This hinders interpretability and may overwhelm the viewer, especially when trying to distinguish minor categories.

4.Small values are hard to perceive
Thin slices representing small percentages are difficult to notice, which undermines their statistical importance. This falls under excessive decoration and misrepresentation, as it reduces graphical integrity.

Improved version:



Above is the improved version of the pie chart. Instead of pie chart, we use horizontal bar chart. This bar chart improves clarity by using aligned bars, making comparisons much easier than in a pie chart. The categories are sorted by size, allowing the viewer to quickly identify the most and least visited platforms. Gridlines and a clear axis support accurate estimation, and the overall layout avoids the clutter seen in the original pie chart.

Note: This improved version is also sourced from the article 'Why you shouldn't use pie charts'.

Part B (Time Series Data):

I. Introduction:

Hyde Park is a racially integrated neighbourhood with a high crime rate. To investigate patterns in criminal activity, we analyse data on reported purse snatchings in Hyde Park from January 1988 to September 1993, recorded every 28 day.

We explore the trend and seasonal structure of the data using stl decomposition and apply month grouping to simplify seasonal modelling. Residual structure is assessed through the "Ident" function, and an AR(2) model is fitted to capture remaining autocorrelation. Diagnostic checks confirm the adequacy of the final model.

II. Analysis:

1. We firstly convert the monthly purse snatching counts into a time series object 'z',and plot this time series data. As shown in Figure 1, the number of incidents fluctuates over time with a visible upward spike around 1990–1991, suggesting a potential trend. There is also a hint of seasonality, as similar patterns seem to repeat across years. These features motivate further decomposition and modelling of trend and seasonal components

.

To assess the need for variance-stabilizing transformation, we examined the log and square root transformations alongside the raw series (Figure 2). Although some degree of heteroscedasticity is present, the variance appears relatively stable over time, and neither transformation provided significant improvement in variance stabilization or interpretability. Therefore, we consider the series to be approximately homoscedastic and proceed with the **untransformed data** for subsequent modelling.

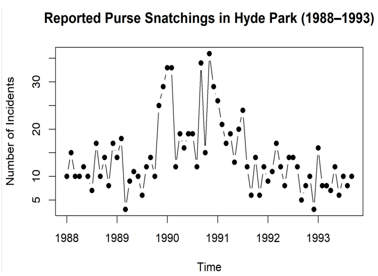


Figure1

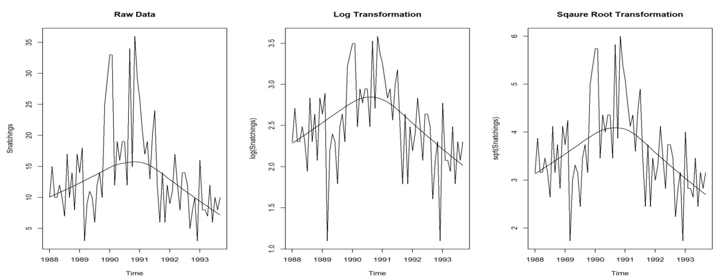


Figure 2

2. We apply a lowess smoother to the raw snatching data, which suggested a curved pattern in the trend, motivating the use of a polynomial model(Figure3).To determine the most suitable polynomial degree, we compared linear, quadratic, and cubic trend fits using robust regression (rlm) in Figure4. For the cubic model, we centred the time variable around 1990 (x <-time(z) - 1990) to reduce multicollinearity.The linear trend underestimated the curvature of the data, while the cubic model, despite fitting more closely, introduced unnecessary complexity. The **quadratic trend** provided a good balance between flexibility and simplicity, and was therefore selected as the final trend model.

The residuals from the quadratic trend show a recurring seasonal pattern in Figure5. Therefore, the next step is to characterise the seasonal effect using monthly indicators

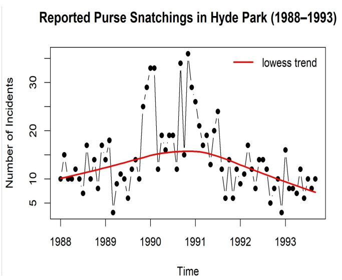


Figure3

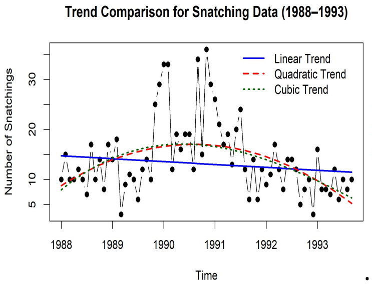


Figure4

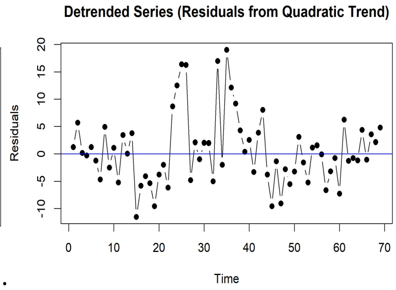


Figure5

3. To model the clear seasonal patterns shown in the stl decomposition and monthplot (Figures6–8), we grouped months based on both average seasonal levels and intra-month trends. The goal was to reduce the number of seasonal parameters while preserving explanatory power.

①Group 1 (March, April, May, June, July, August, September, October): These months exhibit relatively low seasonal values in the monthplot, with generally flat or mild trends across years. While there are some minor deviations—for instance, October shows a slight dip in earlier years and September displays a sharp peak in 1990—these differences are not persistent. The overall magnitude and variability of these months are sufficiently similar to justify grouping them together. We therefore designate **Group 1 as the baseline category for seasonal indicators**.

②Group 2(November, December, January, February): These 4 months have high seasonal values and noticeable within-month variability, as evident in both the stl and the seasonal subseries plot(Figure7,8). Their patterns also share a common rising-and-falling trend, making them suitable for grouping together. **Group2 is assigned an interaction term** with time, as its seasonal pattern differs markedly from that of the baseline group (Group 1), particularly in its time trend.

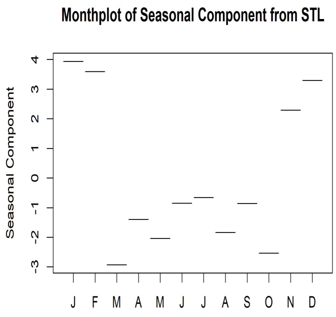


Figure6

Figure7

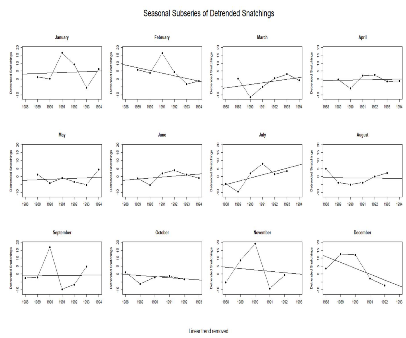


Figure8

This grouping strikes a balance between model simplicity and seasonal interpretability, helping to avoid overfitting while still capturing the key temporal structure in the data. Using this grouping, we created a factor variable v that encodes the month groupings, and fitted a regression model including time, group indicators, and their interaction to assess the seasonality more formally.

v <- Factor(c(rep(c(2,2,1,1,1,1,1,1,1,1,2,2), 5),2,2,1,1,1,1,1,1,1))
trend season model: fits <- rlm(z ~ cbind(time(z), v,v[,1]*time(z)))

To evaluate our grouped seasonal structure, we manually decomposed the series (Figure 10). The seasonal and irregular components resemble those from STL (Figure 7), confirming that our grouping captures key seasonal effects. In Figure 9, the red fitted line reflects seasonal shifts, especially during November–February, but large gaps between the fitted values and raw data remain. This is expected, as the fitted values reflect combined effects of trend, seasonality, and irregular components. These residuals show autocorrelation, indicating the need for further modelling.

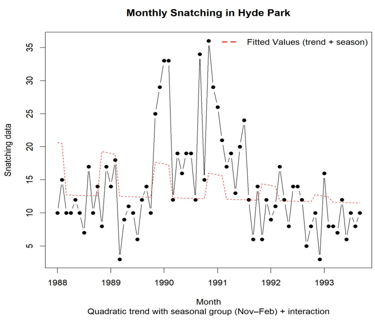


Figure9

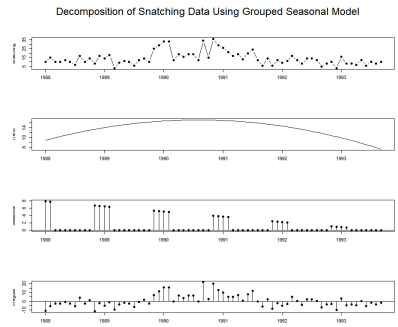


Figure10

4. We examined the residuals from the fitted model (fits) using the Ident function provided in the course. The ACF of the residuals exhibits a gradual decay, indicating that temporal dependence remains and the residuals are not yet white noise. The PACF cuts off after lag 2, suggesting that an AR(2) structure may be appropriate for modeling the remaining autocorrelation.

To address this, we fitted an AR(2) model to the residuals using the Raic function (fin <- Raic(fits$resid), res <- fin$resid[,2]). The residual ACF from this AR model (Figure 12) shows no significant autocorrelation beyond lag 0, and the PACF has no significant spikes, indicating that the irregular component is now approximately stationary and resembles white noise. This confirms that the AR(2) model has effectively removed the remaining serial dependence.

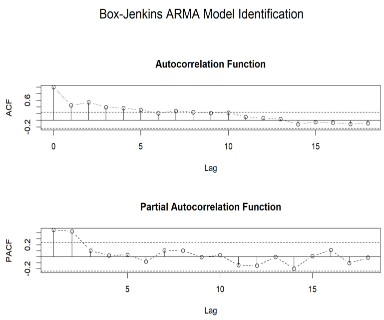


Figure11

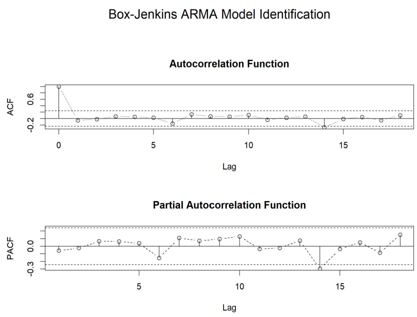


Figure12

We assess the residuals of the fitted AR(2) model using standard diagnostic plots (Figure 13). The residuals show no obvious trend or heteroscedasticity and fluctuate randomly around zero. The Q-Q plot indicates approximate normality, while the residual vs. fitted plot suggests constant variance. These diagnostics confirm that the final model adequately captures the structure in the data. No further trend or autocorrelation modelling is necessary.

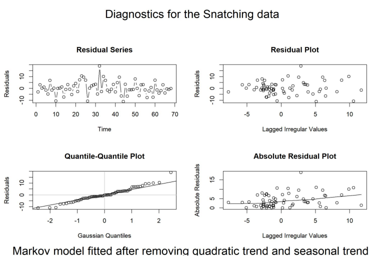


Figure13

```
> fin$coef[1,1]
[1] 0.4300955
> fin$coef[2,2]
[1] 0.4146752
> summary(fits)

Call: rlm(formula = z ~ cbind(time(z), v, v[, 1] * time(z)))
Residuals:
     Min       1Q   Median       3Q      Max
-11.2803  -3.6159  -0.6904   4.3652  21.8284

Coefficients:
                                               Value     Std. Error t value
(Intercept)                                     454.7146 1096.1111     0.4148
cbind(time(z), v, v[, 1] * time(z))time(z)       -0.2223    0.5506    -0.4038
cbind(time(z), v, v[, 1] * time(z))v           2788.1067 2022.2872     1.3787
cbind(time(z), v, v[, 1] * time(z))v[, 1] * time(z)   -1.3985    1.0158    -1.3767

Residual standard error: 5.566 on 65 degrees of freedom
```

Figure14

5. We use fins$coef function and summary(fits) to get our coefficients of final model in Figure14.

$$\mathbf{Final\ model:}\ Y(t) = 454.7146 - 0.2223 \cdot t + (2788.1067 - 1.3985 \cdot t) \cdot I[Nov\text{–}Feb] + X(t)$$

$$X(t) = 0.4301 \cdot X(t-1) + 0.4147 \cdot X(t-2)$$

Where:①Y(t) represents the expected number of monthly snatching incidents in Hyde Park at time t, ②I[Nov–Feb] denotes an indicator function that equals 1 when the month falls in November - February, ③X(t) is the irregular component (modeled as an AR(2) process).

Interpretation of model:

The negative time trend (-0.2223) indicates that purse snatching incidents have been steadily declining over time, suggesting improved public safety in Hyde Park.

The large positive coefficient for November to February (2788.11) confirms these months as the peak snatching season. Although the negative interaction term (-1.3985) implies a slight weakening of this seasonal effect over time, the risk remains significantly elevated. This may reflect increased theft opportunities during colder months and the holiday season. Strengthening police presence during this period would be a strategic crime prevention measure.

Overall, the model effectively captures both seasonal spikes and the long-term decline in incidents. though residual fluctuations highlight the inherent noise and variability of real-world snatching incidents(Figure13). Not only have monthly snatching cases decreased overall, but the gap between peak and off-peak months has also narrowed over time. This suggests that while November to February remain high-risk months, the seasonal concentration of crime has become less pronounced—indicating progress not only in reducing overall crime, but also in smoothing seasonal fluctuations.

III. Conclusion:

This analysis developed a time series model for monthly purse snatching incidents in Hyde Park (1988–1993), incorporating a quadratic trend and grouped seasonal effects. The months from November to February were modeled separately with a time interaction, as this period showed both elevated incident counts and greater temporal variability.

To validate the grouped seasonal structure, stl decomposition was performed before and after applying the month grouping. The seasonal and irregular components remained comparable, confirming that the grouping preserved key seasonal patterns without unnecessary complexity.

Residual autocorrelation was modeled using an AR(2) process, resulting in a stationary, near-white noise irregular component. Diagnostic plots supported the robustness and interpretability of the final model.

The negative coefficients on time and the seasonal interaction suggest a long-term decline in incidents and a diminishing seasonal gap, possibly reflecting improved public safety. However, the persistent peak from November to February highlights the need for increased police presence during this period. Future research could explore structural breaks, external covariates such as temperature or holidays, and real-time forecasting to enhance targeted crime prevention efforts.

Appendix:

```
load(file = "assign3data.RData")

library(MASS)

# The data starts in January 1988, with 12 observations per year (every 28 days)

z <- ts(as.vector(t(snatch)), start = c(1988, 1), frequency = 12)

# Plot the raw time series

plot(z, type = "b", pch = 16,

     main = "Reported Purse Snatchings in Hyde Park (1988–1993)",

     xlab = "Time", ylab = "Number of Incidents")

par(mfrow = c(1, 3))    # Set up a 2x2 plot layout

plot(z, main = "Raw Data", ylab = "Snatchings")

lines(lowess(as.vector(time(z)),z),type='l')

plot(log(z), main = "Log Transformation", ylab = "log(Snatchings)")

lines(lowess(as.vector(time(z)),log(z)),type='l')

plot(sqrt(z), main = "Sqaure Root Transformation", ylab = "sqrt(Snatchings)")

lines(lowess(as.vector(time(z)),sqrt(z)),type='l')

dev.off()

# Plot the raw snatchings time series

plot(z, type = "b", pch = 16,

     main = "Reported Purse Snatchings in Hyde Park (1988–1993)",

     xlab = "Time", ylab = "Number of Incidents")

lines(lowess(time(z), z), col = "red", lwd = 2)

legend("topright", legend = "lowess trend", col = "red", lty = 1, lwd = 2,bty = "n")


#筛选用什么 polynomial

# Linear trend: use time(z) directly

fit_lin <- rlm(z ~ time(z))


# Quadratic trend: still no need to center

fit_quad <- rlm(z ~ cbind(time(z), time(z)^2))


# Cubic trend: center time to reduce collinearity

x <- time(z) - 1990

fit_cubic <- rlm(z ~ cbind(x, x^2, x^3))


# Plot comparison

plot(z, type = "b", pch = 16, col = "black",

     main = "Trend Comparison for Snatching Data (1988–1993)",

     xlab = "Time", ylab = "Number of Snatchings")


lines(as.vector(time(z)), fitted(fit_lin), col = "blue", lty = 1, lwd = 2)

lines(as.vector(time(z)), fitted(fit_quad), col = "red", lty = 2, lwd = 2)

lines(as.vector(time(z)), fitted(fit_cubic), col = "darkgreen", lty = 3, lwd = 2)
```

```
legend("topright",
       legend = c("Linear Trend", "Quadratic Trend", "Cubic Trend"),
       col = c("blue", "red", "darkgreen"),
       lty = 1:3, lwd = 2, bty = "n")
#choose quadratic
trendfit<-fit_quad

#seasonality
plot(trendfit$resid,
     type = "b", pch = 16,
     main = "Detrended Series (Residuals from Quadratic Trend)",
     xlab = "Time", ylab = "Residuals")
abline(h = 0, col = "blue")

detrended <- ts(trendfit$resid, frequency = 12, start = start(z))

# Step 1: STL decomposition
h <- stl(z, s.window = "periodic")

# Step 2: Plot decomposition
plot(h,lwd=2)
title("STL Decomposition of Purse Snatching Series")

# Step 3: Monthplot of the seasonal component only
monthplot(h,
          ylab = "Seasonal Component",
          main = "Monthplot of Seasonal Component from STL")

# Create a month vector from the time series
v2 <- outer(cycle(z), 1:12, "==") * 1
v1 <- v2
apply(v1, 2, sum)
months <- c("January","February","March","April","May","June","July","August",
            "September","October","November","December")

par(mfrow = c(3, 4), oma = c(6, 0, 6, 0))

for (i in 1:12){
  yvals <- z[v1[,i]==1]
  n <- length(yvals)
  xvals <- 1:n

  if(i <= 6){
    yvals <- c(NA, yvals)
```

```
    xvals <- 1:(n+1)
  } else {
    yvals <- c(yvals, NA)
    xvals <- 1:(n+1)
  }

  plot(xvals, yvals,
       type = "b", pch = 16, axes = FALSE,
       ylim = range(z), xlim = c(1, max(xvals)),
       main = months[i],
       ylab = "Snatchings", xlab = "")

  abline(rlm(yvals ~ xvals))

  box()
  axis(2)
  axis(1, at = xvals, labels = 1988:(1988 + length(xvals) - 1))
}
mtext("Seasonal Subseries Plots of Snatching Data",
      side = 3, line = 2, outer = TRUE, cex = 1.5)

detrended <- ts(trendfit$residuals, frequency = 12, start = c(1988, 1))
for (i in 1:12){
  yvals <- detrended[v1[,i] == 1]
  n <- length(yvals)
  xvals <- 1:n

  # Padding with NA to align (same logic as before)
  if(i <= 6){
    yvals <- c(NA, yvals)
    xvals <- 1:(n+1)
  } else {
    yvals <- c(yvals, NA)
    xvals <- 1:(n+1)
  }

  plot(xvals, yvals,
       type = "b", pch = 16, axes = FALSE,
       ylim = range(detrended), xlim = c(1, max(xvals)),
       main = months[i],
       ylab = "Detrended Snatchings", xlab = "")

  abline(rlm(yvals ~ xvals))    # robust linear trend on residuals
  box()
```

```
    axis(2)
    axis(1, at = xvals, labels = 1988 + (0:(length(xvals)-1)))
}

# Add overall title
mtext("Seasonal Subseries of Detrended Snatchings", side = 3, line = 2, outer = TRUE, cex = 1.5)
mtext("Linear trend removed ", side = 1, line = 2, outer = TRUE)

# Define Factor() to create dummy variable matrix (excluding the baseline level)
Factor <- function(a) {
    code <- factor(a)
    nlev <- length(levels(code))
    x <- matrix(0, length(a), nlev - 1)
    dimnames(x) <- list(NULL, levels(code)[2:nlev])
    for (i in 2:nlev) {
        x[code == levels(code)[i], i - 1] <- 1
    }
    return(x)
}
#build new season vector
v <- Factor(c(
    rep(c(2,2,1,1,1,1,1,1,1,1,2,2), 5),
    2,2,1,1,1,1,1,1,1
))

# Fit model with quadratic trend and one seasonal group (v2) plus interaction
fits <- rlm(z ~ cbind(time(z), v, v[,1] * time(z)))

# Plot fitted trend + season vs raw data
dev.off()
plot(z, xlab = "Month", ylab = "Snatching data", type = "b", pch = 16,
     main = "Monthly Snatching in Hyde Park",
     sub = "Quadratic trend with seasonal group (Nov–Feb) + interaction")
lines(as.vector(time(z)), fitted(fits), lty = 2, col = "red")
legend("topright",
       legend = "Fitted Values (trend + season)",
       col = "red",
       lty = 2,
       lwd = 2,
       bty = "n")

# Seasonal decomposition components
season <- c(
    rep(c(2,2,1,1,1,1,1,1,1,1,2,2), 5),
```

```
    2,2,1,1,1,1,1,1,1
)

# Coefficients
b <- fits$coef

# Only one group effect (Group 2)
ss <- rep(0, 2)
ss[2] <- b[3]

# Build seasonal component with interaction term for Group 2
seasonal <- ss[season] +
    ifelse(season == 2, b[4], 0) * time(z)

# Plot decomposition: original, trend, seasonal, irregular
par(mfrow = c(4, 1), oma = c(0, 0, 3, 0))

# Original series
plot(1:length(z), z, type = "b", pch = 16, axes = FALSE,
        ylab = "Snatching", xlab = "")
box()
axis(2)
axis(1, at = seq(1, length(z), 12), labels = 1988:1993)

# Trend
trend <- fitted(trendfit)
plot(1:length(z), trend, type = "l", axes = FALSE,
        ylab = "Trend", xlab = "")
box()
axis(2)
axis(1, at = seq(1, length(z), 12), labels = 1988:1993)

# Seasonal
plot(1:length(z), seasonal, type = "p", pch = 16, axes = FALSE,
        ylab = "Seasonal", xlab = "")
abline(h = 0)
segments(1:length(z), 0, 1:length(z), seasonal)
box()
axis(2)
axis(1, at = seq(1, length(z), 12), labels = 1988:1993)

# Irregular
plot(1:length(z), fits$resid, type = "p", pch = 16, axes = FALSE,
        ylab = "Irregular", xlab = "")
```

```
abline(h = 0)
segments(1:length(z), 0, 1:length(z), fits$resid)
box()
axis(2)
axis(1, at = seq(1, length(z), 12), labels = 1988:1993)

# Title
mtext("Decomposition of Snatching Data Using Grouped Seasonal Model",
      side = 3, outer = TRUE, cex = 1.5)

summary(fits)
dev.off()
plot(trendfit$resid,xlab="Month",ylab="residual if trendfit",type='b',pch=16,
     main="Detrended Monthly Snatching in Hyde Park")
plot(fits$residuals,
     xlab = "Month",
     ylab = "Residual of fits",
     type = "b",
     pch = 16,
     main = "Detrended and Deseasonalised Monthly Snatching in Hyde Park",
     sub = 'Quadratic trend and grouped seasonal')
abline(h=0)

# Box-Jenkins Identification for the irregular series
Ident<-function(x, data = NULL, sub = NULL)
{
  u <- acf(x, plot = F)
  v <- acf(x, type = "partial", plot = F)
  c1 <- 2./sqrt(u$n.used)
  oldpars <- par()
  oldpars$pin <- c(7.256, 5.216)
  par(mfrow = c(2., 1.), oma = c(3., 0., 4., 0.))
  plot(u$lag, u$acf, type = "b", lty = 3., ylab = "ACF", xlab = "Lag",
       ylim = range(u$acf, -c1, c1), main =
         "Autocorrelation Function")
  segments(u$lag, 0., u$lag, u$acf)
  box(lty = 1.)
  abline(h = c(-c1, c1), lty = 2.)
  abline(h = 0., lty = 1.)
  plot(v$lag, v$acf, type = "b", lty = 2., ylab = "PACF", xlab = "Lag",
       ylim = range(v$acf, -c1, c1), main =
         "Partial Autocorrelation Function")
  segments(v$lag, 0., v$lag, v$acf)
  box(lty = 1.)
```

```
    abline(h = c(-c1, c1), lty = 2.)
    abline(h = 0., lty = 1.)
    if(is.null(data))
        mtext("Box-Jenkins ARMA Model Identification", 3., 2., outer
                = T, cex = 1.5)
    else mtext(paste("Box-Jenkins ARMA Model Identification for", data),
                3., 2., outer = T, cex = 1.5)
    if(!is.null(sub))
        mtext(paste(sub), 1., 2., outer = T)
    invisible()
}
Ident(fits$resid)


.startval<-function (y, Z, tol)
{type <- FALSE          # hard-coded, TRUE as proposed in [MarT82b], [StoDut87]
    #                         FALSE as proposed in [MarZ78]
    ar.ls <- lm.fit(Z, y, tol)
    phi <- ar.ls$coefficients
    if (type) {
        s <- sqrt(sum((ar.ls$residuals)^2)/ar.ls$df.residual)
    } else {
        s <- mad(ar.ls$residuals)
    }
    return(list(phi=phi, s=s))
}
psiHuber<-function (t, k=1.345, rho=FALSE)
{
    if (!rho) {
        pmin(k, pmax(-k, t))
    } else {
        r <- abs(t)
        i <- r < k
        r[i] <- r[i]^2/2
        r[!i] <- k * (r[!i] - k/2)
        r
    }
}
.psi<-function (type)
{

    switch(type,
              Ident = get("psiLS", mode="function"),
              Huber = get("psiHuber", mode="function"),
```

```
             Tukey = get("psiTukey", mode="function"),
             Hampel = get("psiHampel", mode="function"))
}
.weights<-function (r, s, u, v, psi1, ...)
{psi <- .psi(psi1)
  n <- length(r)
  w <- rep(NA, n)
  for (i in 1:n) {
    if (r[i] == 0) {
      if (u[i] != 0) {
        w[i] <- v[i]/u[i]
      } else {
        w[i] <- 1
      }
    } else if (u[i] != 0) {
      dummy <- r[i]/s
      w[i] <- v[i]*psi(dummy/u[i], ...)/dummy
    } else if (psi1 == "Ident") {
      w[i] <- 1
    } else {
      w[i] <- 0
    }
  }
  return(w)
}
.Weights<-function (p, Z, invCp, type, psi2, c)
{

  psi <- .psi(psi2)
  d <- sqrt(diag(Z%*%invCp%*%t(Z))/p)
  if (psi2 == "Huber") {
    v <- psi(d, k=c)/d
  } else if (psi2 == "Tukey") {
    v <- psi(d, c=c)/d
  } else if (psi2 == "Ident") {
    v <- rep(1, nrow(Z))
  } else {
    warning("error in function '.Weights': psi function ", psi2,
              "not defined \n")
  }
  if (type=="Mallows") {
    u <- rep(1, length(v))
  } else if (type=="Schweppe") {
    u <- v
```

```
    } else {
        warning("error in function '.Weights': wrong GM-estimates type \n")
    }
    return(list(u=u, v=v))
}
.invCp<-function (p, s, Phi)
{

    if (p > 1) {
        M1 <- matrix(rep(rev(s), p), p)
        M2 <- cbind(rep(0, (p-1)), (-1)*Phi)
        M2 <- rbind(M2, rep(0, p))
        diag(M2) <- 1
        Ap <- (1/M1)*M2
        invCp <- t(Ap)%*%Ap
    } else {
        invCp <- 1/s[1]^2
    }
    return(invCp)
}
.ARmodel<-function (x, p)
{

    n <- length(x)
    y <- x[(p+1):n]
    Z <- embed(x, p)[1:(n-p), ]
    return(list(y=y, Z=as.matrix(Z)))
}
.BH<-function (k=1.345)
{-2*k*dnorm(k) + 2*pnorm(k) + 2*k^2*(1-pnorm(k)) -1
}
.IWLS<-function (y, Z, phi.ini, s.ini, u, v, psi1, niter, tol, ...)
{stop1 <- sqrt(diag(solve(t(Z)%*%Z)))
    B <- NA
    phi <- phi.ini
    s.new <- s.ini
    iter <- 0
    psi <- .psi(psi1)
    if (psi1=="Huber") {
        B <- .BH(...)
    } else {
        B <- .BB(...)
    }
```

```
  while (iter < niter) {
    iter <- iter + 1
    r <- y - Z%*%phi
    s.old <- s.new
    s2 <- (1/(B*sum(u*v)))*sum(u*v*(psi(r/(u*s.old), ...))^2)*s.old^2
    s.new <- sqrt(s2)
    w <- .weights(r, s.new, u, v, psi1, ...)
    ar.wls <- lm.wfit(Z, y, w, tol)
    tau <- ar.wls$coefficients - phi
    omega <- 1          # hard-coded, 0 < omega < 2, as proposed in [StoDut87]
    phi <- phi + omega*tau
    stop2 <- abs(c(s.old - s.new, omega*tau)) < tol*s.new*c(1, stop1)
    if (sum(stop2) == length(stop2)) break
  }
  return(list(phi=phi, s=s.new, w=w, B=B, niter=iter))
}
arGM<-function (x, order=1,
          chr=1.5, iterh=maxiter, cbr=5.0, iterb=maxiter,
          psi2="Tukey", c=4.0, type="Mallows",
          k=1.5, maxiter=100, tol=1e-08, equal.LS=FALSE,...)
{

  s <- c()
  Phi <- matrix()
  w <- NA
  BH <- BB <- NA
  niterh <- niterb <- niter.testing <- NA

  ##    Centering:

  ## x.huber <- HuberM(x, ...)          # as proposed in [StoDut87]
  x.huber <- hubers(x)          # as proposed in [MarZ78], [Mart80]
  x <- x - x.huber$mu
  sx <- x.huber$s

  ##    Main:

  for (p in 1:order) {
    ARmodel <- .ARmodel(x, p)
    y <- ARmodel$y
    Z <- ARmodel$Z
    invCp <- .invCp(p, c(sx, s), Phi)
    Weights <- .Weights(p, Z, invCp, type, psi2, c)
    u <- Weights$u
```

```
    v <- Weights$v
    startval <- .startval(y, Z, tol)
    phi <- startval$phi
    s[p] <- startval$s
    if (equal.LS) {          # for testing purpose only
        psi1 <- "Ident"
        niter <- maxiter
        IWLS <- .IWLS(y, Z, phi, s[p], u, v, psi1, niter, tol)
        phi <- IWLS$phi
        w <- IWLS$w
        niter.testing <- IWLS$niter
    } else {
        if ((iterh > 0) & (is.numeric(iterh))) {
            psi1 <- "Huber"
            niter <- iterh
            IWLS <- .IWLS(y, Z, phi, s[p], u, v, psi1, niter, tol, k=chr)
            phi <- IWLS$phi
            s[p] <- IWLS$s
            w <- IWLS$w
            BH <- IWLS$B
            niterh <- IWLS$niter
        }
        if ((iterb > 0) & (is.numeric(iterb))) {
            psi1 <- "Tukey"
            niter <- iterb
            IWLS <- .IWLS(y, Z, phi, s[p], u, v, psi1, niter, tol, c=cbr)
            phi <- IWLS$phi
            s[p] <- IWLS$s
            w <- IWLS$w
            BB <- IWLS$B
            niterb <- IWLS$niter
        }
    }
    if (p > 1) {
        Phi <- cbind(rep(0, (p-1)), Phi)
        Phi <- rbind(phi, Phi)
    } else {
        Phi <- as.matrix(phi)
    }
}
Cx <- solve(invCp)

return(list(ar=phi, sinnov=s, Cx=Cx, mu=x.huber$mu, sx=sx, u=u, v=v, w=w,
            BH=BH, BB=BB,
```

```
                    niterh=niterh, niterb=niterb, niter.testing=niter.testing))

}
Raic<-function(x, order = 10.)
{
   n <- length(x)
   z <- matrix(0., n - 1., order)
   for(k in 1.:order)
      z[(order + 1. - k):(n - 1.), k] <- x[1.:(n - order - 1. + k)]
   res <- matrix(0., n - 1., order)
   mu <- rep(0., order)
   svec <- mu
   ar <- matrix(0., order, order)
   y <- arGM(x, order, chr = 1.345, psi2="Huber", iterb=0)
   mu[order] <- y$mu
   svec[order] <- y$sinnov[order]
   ar[, order] <- y$ar
   s <- y$sinnov[order]
   res[, order] <- x[2.:n] - rep(y$mu, n - 1.) - z[, order:1.] %*% y$ar
   for(k in 1.:(order - 1.)) {
      y <- arGM(x, k,    chr = 1.345, psi2="Huber", iterb=0)
      mu[k] <- y$mu
      svec[k] <- y$sinnov[k]
      ar[1.:k, k] <- y$ar
      s <- y$sinnov[k]
      res[, k] <- x[2.:n] - rep(y$mu, n - 1.) - as.matrix(z[, order:
                                                                  (order - k + 1.)]) %*% as.vector(y$ar)
   }
   list(k = c(1.:order), resid = res, coef = ar, mu = mu,
          sd = svec)
}
fin <- Raic(fits$resid)

res <- fin$resid[,2]
fv <- fits$resid[-2] - res
Ident(res)

#dignostics plot
par(mfcol=c(2,2),oma = c(6,0,6,0))

plot(1:(length(z)-1),res,type="b",
       main="Residual Series",
       xlab="Time",
       ylab="Residuals")
```

```
qqnorm(res,
       main="Quantile-Quantile Plot",
       xlab="Gaussian Quantiles",
       ylab="Residuals")
qqline(res);abline(h=0,v=0,col="grey")

plot(fv,res,
     main="Residual Plot",
     xlab="Lagged Irregular Values",
     ylab="Residuals")

plot(fv,abs(res),
     main="Absolute Residual Plot",
     xlab="Lagged Irregular Values",
     ylab="Absolute Residuals")
lines(lowess(fv,abs(res)))

mtext("Diagnostics for the Snatching data",
      side=3,line=2,outer=T,cex=1.5)
mtext("Markov model fitted after removing quadratic trend and seasonal trend
      ",
      side=1,line=2,outer=T,cex=1.5)
fin$coef[1,1]
fin$coef[2,2]
summary(fits)
```