Nonparametric Statistics IV

Mini-project Epiphany term

Scott Phillips

2025-04-29

Introduction

This report uses the "openair" R package, which provides a suite of function to access the "openair" database. This database is a collection of air pollution measurements from across the UK, in particular it tracks the concentrations of harmful chemicals and pollutants over time. Using this data, we will investigate the mean concentration of harmful chemicals in the UK's airspace, denoted p_m, from 2002 to 2022.

The goal of this report is to construct and compare analytic and bootstrap-based confidence intervals for the local linear regression estimator of p_m. This will be done by first fitting the estimator to the data, then calculating the optimal bandwidth, and finally constructing the confidence intervals. This approach will demonstrate the application of each methodology and allow for a direct comparison of their results.

Data Preparation

```
library(openair)
library(lubridate)
library(SemiPar)
library(locpol)

sites <- importMeta(source = 'aurn')
#Access the UK urban and rural network

sitestotal <- sites$code
#Filter for the site codes

UK_poll_data <- importUKAQ(site = sitestotal, year = 2002:2022, data_type = '')
#Access all data between 2002 and 2022

UK_poll_data_omit <- na.omit(UK_poll_data)

UK_poll_data$day <- as.Date(UK_poll_data$date)
#Create a new variable in as.date format for use in aggregate function

daily_mean <- aggregate(pm2.5 ~ day, data = UK_poll_data,
                        FUN = function(x) mean(x, na.rm = TRUE))
#the data is recorded hourly, average this over each day and then find the mean
#over all the sites in the UK
```

```r
daily_mean$decimal.year <- decimal_date(daily_mean$day)
#turn as.date format into numeric for use in functions

Y <- daily_mean$pm2.5
X <- daily_mean$decimal.year
X_centered <- X - mean(X)
```

Local Linear Estimator

We commence our analysis of this dataset by first constructing the local linear estimator $\hat{m}(x)$ as shown in section 4.2.2. It takes the form:

$$\hat{m}_{LL}(x) = \frac{\sum_{i=1}^{n} v_i(x) y_i}{\sum_{i=1}^{n} v_i(x)}.$$

Where we have that

$$v_i(x) = K(\frac{x - x_i}{h})(S_{n,2} - (x - x_i)S_{n,1}),$$

with

$$S_{n,j} = \sum_{i=1}^{n} K(\frac{x - x_i}{h})(x - x_i)^j,$$

We will use a Gaussian kernel due to the noise that is prevalent through the dataset and this can then be implemented in R as follows using:

```r
my.kernel <- function(u){
  u <- dnorm(u)
}

Sn<- function(xdat, x, h, j){
  snj<-  sum(my.kernel((xdat-x)/h)*(xdat-x)^j)
  return(snj)
}

vix<-function(xdat, x, h){
  my.kernel((xdat-x)/h)*( Sn(xdat,x,h,2)-(xdat-x)*Sn(xdat,x,h,1) )
}

m.est <- function(xdat, ydat, xgrid = xdat, h){
  G <- length(xgrid)
  est <- rep(0, G)
  for(j in 1:G){
    est[j] <- sum(ydat*vix(xdat, xgrid[j], h))/sum(vix(xdat, xgrid[j],h))
  }
  return(as.data.frame(cbind(xgrid, est)))
}
```

Optimal Bandwidth

With the estimator constructed, we can now address the issue of finding the optimal bandwidth for the local linear estimator. The optimal bandwidth can be found as follows:

$$h_{opt} = \left[\frac{\nu_0}{\mu_2^2}\frac{\int \sigma^2(x)/f(x)dx}{\int m''(x)^2 dx}\right]^{1/5} n^{-1/5}.$$

The problem with this bandwidth is it involves the unknown quantities $\sigma^2(x), f(x)$ and $m''(x)$. These quantities can be estimated through the use of pilot estimators. One such application of this is the Fan and Gijbels (1996) "Rule-of-thumb" bandwidth method. It can be seen as follows:

1. Estimate $\hat{m}(x)$ globally by a polynomial of degree 4
$$\hat{m}(x) = \hat{\alpha}_0 + \hat{\alpha}_1 x + \hat{\alpha}_2 x^2 + \hat{\alpha}_3 x^3 + \hat{\alpha}_4 x^4.$$
2. Estimate a constant error standard deviation σ from the residuals of the fitted model
$$\hat{\sigma^2} = \frac{1}{n-5}\sum_{i=1}^{n}(y_i - \hat{m}(x_i))^2.$$
3. Compute
$$\hat{m}''(x) = 2\hat{\alpha}_2 + 6\hat{\alpha}_3 x + 12\hat{\alpha}_4 x^2.$$

This then results in an optimal bandwidth of the form

$$h_{ROT} = \left[\frac{\nu_0}{\mu_2^2}\frac{\hat{\sigma}^2}{\sum_{i=1}^{n}\hat{m}''(x)^2}\right]^{(1/5)} n^{(-1/5)}.$$

Instead of manually implementing this function, we will use the function 'thumbBw' in the 'locpol' package. In reference to our dataset, it is necessary to center the time variable due to it resulting in infinities in the calculation. This can be implemented as follows:

```
poll_h_opt <- thumbBw(X_centered, Y, deg = 1, kernel = locpol::gaussK)
poll_h_opt
```

```
## [1] 0.5798828
```

Analytic Confidence Intervals

With the regression estimator and the optimal bandwidth found, we can now derive the analytic confidence intervals in a similar way to what is shown in section 3.3.9. We start by applying the central limit theorem:

$$\frac{\hat{m}_h(x) - \mathbb{E}(\hat{m}_h(x))}{Var(\hat{m}_h(x))} \xrightarrow{d} N(0,1).$$

Using the fact that $\mathbb{E}(\hat{m}_h(x)) = m_h(x) + Bias(\hat{m}_h(x))$, we need to find the bias and variance of the estimator. From lectures we can find these to be:

$$Bias(\hat{m}_{LL}(x)) = \frac{1}{2}h^2\mu_2 m''(x) + O(h^3),$$

$$Var(\hat{m}_{LL}(x)) = \frac{\nu_0\sigma^2(x)}{nhf(x)} + o\left(\frac{1}{nh}\right).$$

Applying these to the previous formula results in:

$$\frac{\hat{m}_h(x) - m(x) - \frac{h^2}{2}\mu_2 m''(x) + O(h^3)}{\sqrt{\frac{\nu_0 \sigma^2(x)}{nhf(x)} + o\left(\frac{1}{nh}\right)}} \xrightarrow{d} N(0,1).$$

This can then be rewritten as:

$$\sqrt{nh}\frac{\hat{m}_h(x) - m(x) - \frac{h^2}{2}\mu_2 m''(x) + O(h^3)}{\sqrt{\frac{\nu_0 \sigma^2(x)}{f(x)} + o(1)}} \xrightarrow{d} N(0,1).$$

From this we can now discard the $o(1)$ under the assymptotic assumptions and we can say:

$$\sqrt{nh}\left(\hat{m}_{LL}(x) - m_{LL}(x) - \frac{h^2}{2}\mu_2 m''(x) + O(h^3)\right) \xrightarrow{d} N(0, \frac{\nu_0 \sigma^2(x)}{f(x)}).$$

Now we would like to find a confidence interval of the form:

$$\sqrt{nh}\left(\hat{m}_{LL}(x) - m_{LL}(x)\right) \xrightarrow{d} N(0, \frac{\nu_0 \sigma^2(x)}{f(x)}).$$

In order to achieve this confidence interval, we require the bias terms to get asymptotically erased when multiplied with $\sqrt{nh}$.

1. For the $O(h^3)$ term, taking $h_{opt} = c \cdot n^{-1/5}$ we get

$$\sqrt{nh}O(h_{opt}^3) = O(n^{1/2}h_{opt}^{7/2}) = O(n^{1/2}(cn^{-1/5})^{7/2}) = n^{-1/5} \to 0.$$

2. For the $O(h^2)$ term,

$$\sqrt{nh}O(h_{opt}^2) = O(n^{1/2}h_{opt}^{5/2}) = O(n^{1/2}(cn^{-1/5})^{5/2}) = O(1).$$

We can see that this term does not vanish unless:

$$\sqrt{nh}h^2 = o(1), \quad nh^5 = o(1), \quad h = o(n^{-1/5}).$$

From our previously calculated optimal bandwidth and knowledge of the size of our dataset, we know this bias term does not disappear. However, as shown in section 3.3.9, it can be disregarded if h is small in relation to n as this is effectively ensuring $h = o(n^{-1/5})$. Since we have an optimal bandwidth of 0.5799 and sample size equal to 7670 this stands and we can disregard the bias in our calculations and use the analytic confidence intervals as shown below:

$$\hat{m}_{LL}(x) \pm z_{1-\alpha/2}\sqrt{\frac{\sigma^2(x)\nu_0}{nh\hat{f}_h(x)}}.$$

In this analytic confidence interval, there are more unknown quantities that must be estimated. We know from the definition of the local linear estimator that:

$$\hat{f}_h(x) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x - x_i}{h}).$$

This can be implemented into R as follows:

```r
f.est <- function(xdat, xgrid, h){
  n <- length(xdat)
  g <- length(xgrid)
  est <- rep(0,g)
  for(j in 1:g){
    est[j] <- sum(my.kernel((xdat - xgrid[j])/h))/(n*h)
  }
  return(est)
}
```

It is worth mentioning here that there is no need to calculate a new optimal bandwidth for this kernel. This is due to the response variables bandwidth being made redundant when calculating the local linear estimator as shown in section 4.1.2 and corroborated by Chen and Qin (2002).

Now estimating the unknown quantity $\sigma(x)^2$, we can use the result from Hardle (1990) to do this. In this approximation, the bias of the estimator is ignored. However, as we have shown that bias is not included in our estimation, this is not an issue and will be a very good approximation. The estimation can be found to be:

$$\sigma(\hat{x})^2 = n^{-1} \sum_{i=1}^{n} W_i(x)(Y_i - \hat{m}_{LL}(x))^2,$$

where

$$W_i(x) = \frac{\frac{1}{h} K\left(\frac{x-x_i}{h}\right)}{f(x)}.$$

This can be implemented in R as follows:

```r
sig.est <- function(xdat, ydat, xgrid, h){
  n <- length(xdat)
  g <- length(xgrid)
  est <- rep(0,g)
  mest <- m.est(xdat, ydat, xdat, h)$est
  #Calculate ll estimator over data values
  resid <- (ydat - mest)^2
  #Calculate residuals
  for(i in 1:g){
    weight <- (my.kernel((xdat - xgrid[i])/h))
    weight <- weight/sum(weight)
    #calculate weights
    est[i] <- sum(weight * resid)
  }
  return(est)
}
```

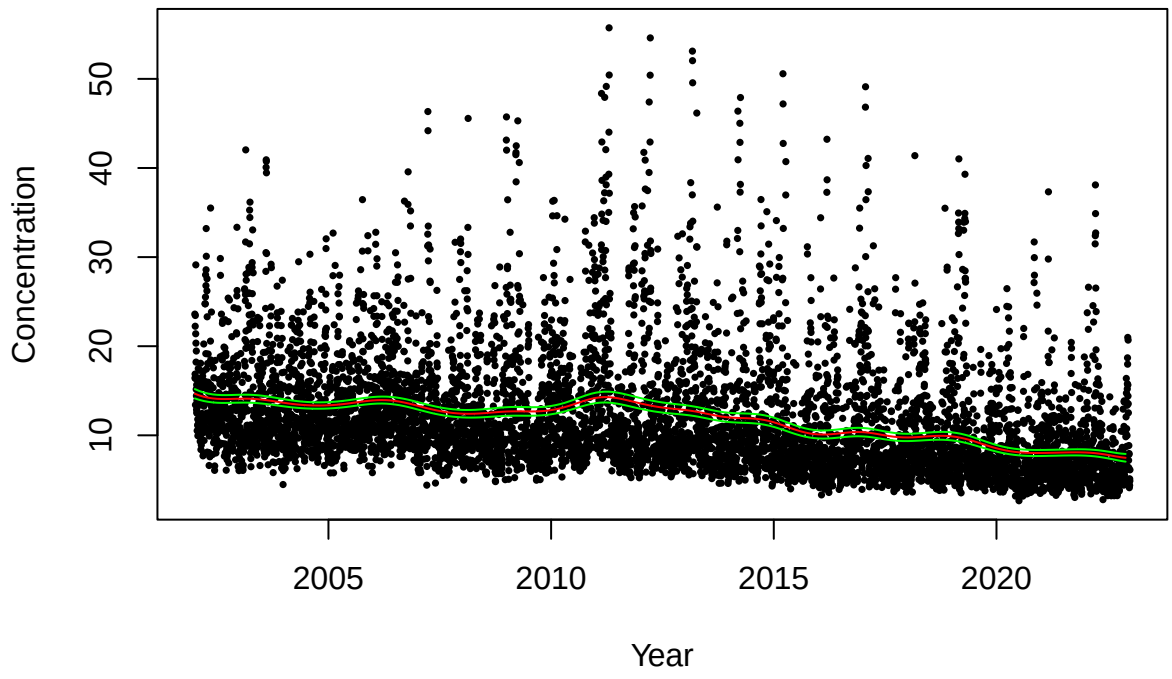
Finally we can implement all of this into a function that calculates the normal confidence interval. In this application we will use a Gaussian kernel as it is efficient when dealing with noisy data like the 'openair' dataset. It can be implemented in R as follows:

```r
CI.est <- function(xdat, ydat, xgrid, h, alpha){
  start <- Sys.time()
  nu0 <- 1/(2*sqrt(pi))
  n <- length(xdat)
  g <- length(xgrid)
  lower <- rep(0, g)
```

```
  upper <- rep(0, g)
  #Calculate m(x), sigma^2(x) and f(x)
  mx <- m.est(xdat, ydat, xgrid, h)
  sigma2 <- sig.est(xdat, ydat, xgrid, h)
  fx <- f.est(xdat, xgrid, h)
  for(i in 1:g){
    upper[i] <- mx$est[i] + qnorm(1-alpha/2) * sqrt((sigma2[i] * nu0)/(n * h * fx[i]))
    lower[i] <- mx$est[i] - qnorm(1-alpha/2) * sqrt((sigma2[i] * nu0)/(n * h * fx[i]))
  }
  end <- Sys.time()
  time <- end-start
  return(c("est"=list(mx$est),"lower"=list(lower),"upper"=list(upper), "time"=list(time)))
}
```

With the groundwork set, we can now apply this to our dataset to find a 95% confidence interval, this results in the following plot where red represents the local linear estimator and green the confidence intervals:

```
pollution.grid <- seq(min(X), max(X), by = 0.1)
pollution.est <- CI.est(X, Y, pollution.grid, poll_h_opt, 0.05)
plot(X, Y, pch = 16, cex = 0.5, xlab = 'Year', ylab = 'Concentration')
lines(pollution.grid, pollution.est$est, col = 'red', lwd = 1)
lines(pollution.grid, pollution.est$lower, col = 'green', lwd = 1)
lines(pollution.grid, pollution.est$upper, col = 'green', lwd = 1)
```



This is clearly a very narrow confidence interval, likely due to the large dataset. Nonetheless it provides

valuable insight into the uncertainty of the kernel density estimation. Now all that is left to do is to consider the mean width of the confidence interval for comparison later, it can be found as follows:

```
analytic_width <- mean(pollution.est$upper - pollution.est$lower)
print(c(analytic_width = analytic_width, time_minutes = as.numeric(pollution.est$time, units = "mins")))
```

```
## analytic_width   time_minutes
##      0.8987572      1.8735790
```

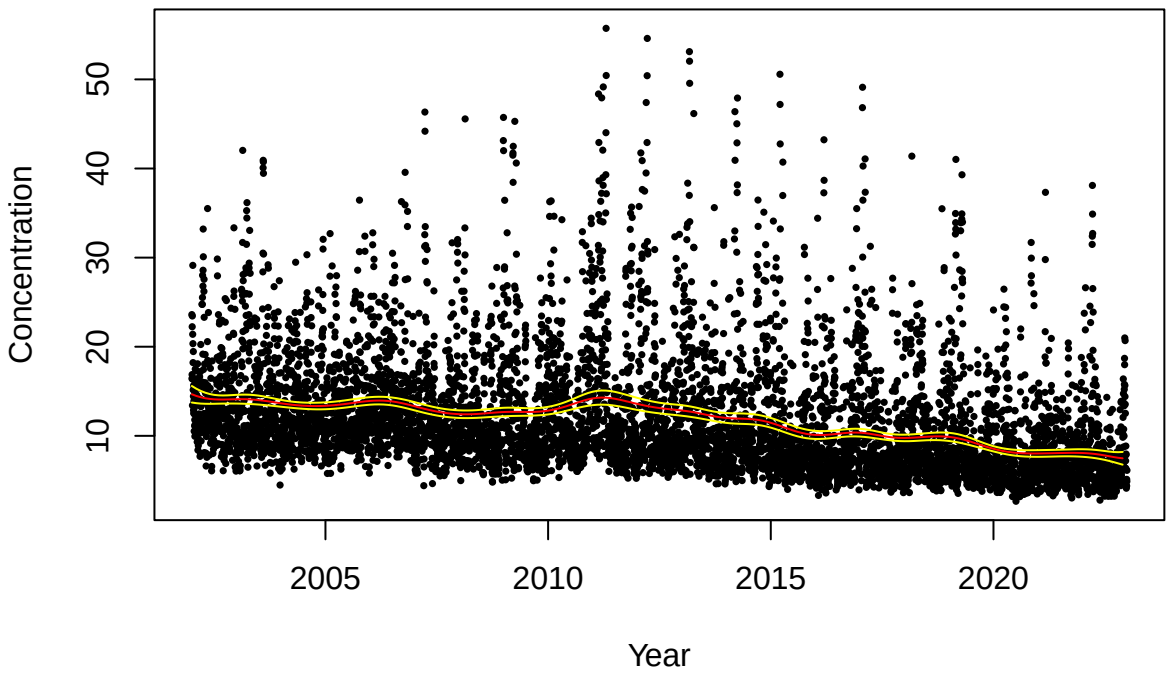
Bootstrap Confidence Intervals

We now move onto the construction of bootstrap confidence intervals, we start with the nonparametric paired bootstrap. This will be done by calculating the local linear estimator $\hat{m}(x)$ over a grid of values of our choosing on B resampled pairs (x_i, y_i) from the dataset of length n. From this we will then calculate the 95% normal confidence intervals. This can be implemented in R as follows:

```
bootstrap_mll <- function(xdat, ydat, xgrid, B, h){
  start <- Sys.time()
  n.boot <- length(ydat)
  m <- length(xgrid)
  est <- matrix(NA, nrow = m, ncol = B)
  for(j in 1:B){
    indicies <- sample(1:n.boot, replace = TRUE)
    x.boot <- xdat[indicies]
    y.boot <- ydat[indicies]
    est[,j] <- m.est(x.boot, y.boot, xgrid, h)$est
    #calculate estimator using resampled data
  }
  end <- Sys.time()
  time <- end-start
  return(list(est, time))
}
set.seed(1)
boot.mll <- bootstrap_mll(X, Y, pollution.grid, 100, poll_h_opt)
mll <- m.est(X, Y, pollution.grid, poll_h_opt)
boot.mll.se <- apply(boot.mll[[1]], 1, sd)

normal.CI <- cbind(mll[,2] - qnorm(0.975) * boot.mll.se,
                   mll[,2] + qnorm(0.975) * boot.mll.se)

plot(X, Y, pch = 16, cex = 0.5, xlab = 'Year', ylab = 'Concentration')
lines(mll, col = 'red', lwd = 1)
lines(y = normal.CI[,1], x = pollution.grid, col = 'yellow', lwd = 1)
lines(y = normal.CI[,2], x = pollution.grid, col = 'yellow', lwd = 1)
```



In this plot the confidence intervals are yellow and estimator is red. This confidence interval is very similar to the one shown before apart from the tails, which in this case are wider, suggesting an increase in uncertainty. This is likely due to the restrictive assumptions placed on the analytical confidence interval that falsely increases our certainty about the kernel in the analytic case. We have chosen to use 100 bootstrap resamples due to the computational burden of computing this over such a large dataset.

We can now calculate the width of the confidence intervals and the computation time as before:

```
paired_width <- mean(normal.CI[,2] - normal.CI[,1])
print(c(paired_width = paired_width, time_minutes = as.numeric(boot.mll[[2]], units = "mins")))
```

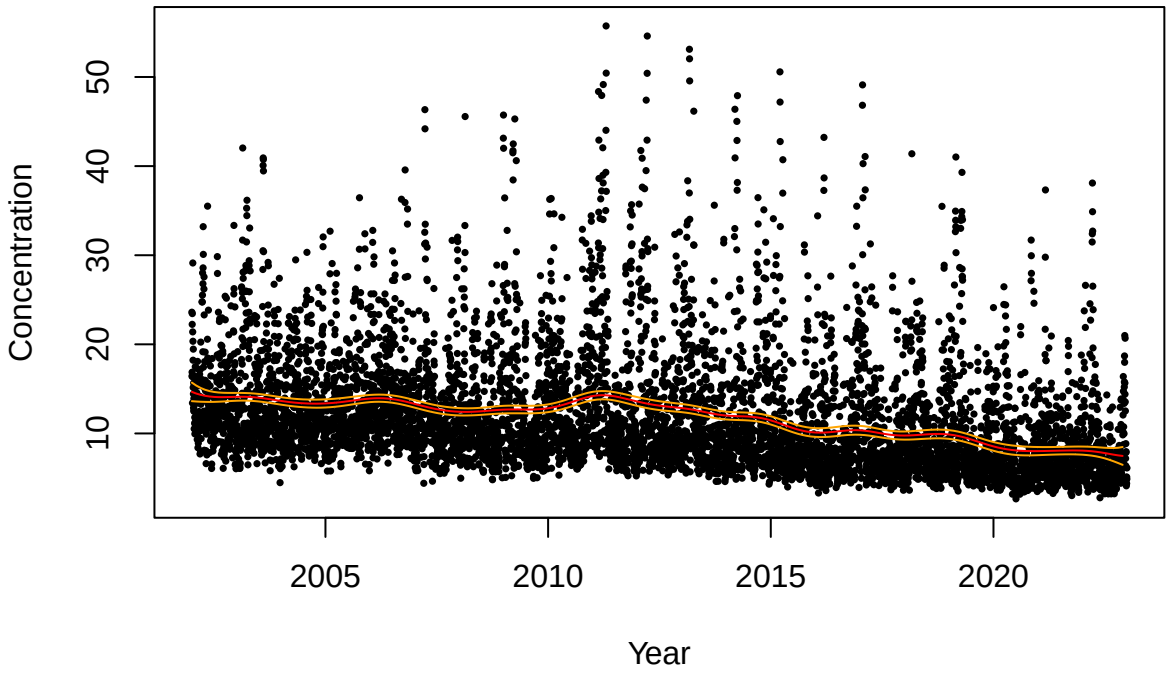
```
## paired_width time_minutes
##    0.9536284    2.6433227
```

We now can move onto the final method of interest, the semiparametric residual bootstrap. This is done by calculating the local linear regression estimator over all the dataset values (x_i, y_i) on the grid x_i. The residuals of this are then calculated, from which they are then resampled B times. From here the bootstrapped y_i are calculated by adding the estimator's values to the resampled residuals. Finally, using these resampled y-values, the local linear estimator is calculated over a grid of values of our choosing and then a normal confidence interval can be constructed from this. This can be implemented as follows:

```
NPBbootstrap_mll <- function(xdat, ydat, xgrid, B, h) {
  start <- Sys.time()
  m <- length(xgrid)
  est <- matrix(NA, nrow = m, ncol = B)
  mx <- m.est(xdat, ydat, xdat, h)[,2]
```

```
  #Calculate ll estimator using original data
  residuals <- ydat - mx
  # Calculate residuals
  for (j in 1:B) {
    eps.boot <- sample(residuals, replace = TRUE)
    y.boot <- mx + eps.boot
    est[,j] <- m.est(xdat, y.boot, xgrid, h)[,2]
    # calculate ll estimator using resampled y values from resampled residual values
  }
  end <- Sys.time()
  time <- end-start
  return(list(est, time))
}
set.seed(1)
NPBboot.mll <- NPBbootstrap_mll(X, Y, pollution.grid, 100, poll_h_opt)
NPBboot.mll.se <- apply(NPBboot.mll[[1]], 1, sd)
NPBnormal.CI <- cbind(mll[,2] - qnorm(0.975) * NPBboot.mll.se,
                      mll[,2] + qnorm(0.975) * NPBboot.mll.se)

plot(X, Y, pch = 16, cex = 0.5, xlab = 'Year', ylab = 'Concentration')
lines(mll, col = 'red', lwd = 1)
lines(y = NPBnormal.CI[,1], x = pollution.grid, col = 'orange', lwd = 1)
lines(y = NPBnormal.CI[,2], x = pollution.grid, col = 'orange', lwd = 1)
```



Once again in this plot the estimator is red and the confidence intervals are orange. This confidence interval is again very similar to the previous intervals and seems reasonable. To conclude our discussion of the

residual bootstrap, we can calculate the width of these confidence intervals and the computation time which is implemented as follows:

```
residual_width <- mean(NPBnormal.CI[,2] - NPBnormal.CI[,1])
print(c(residual_width = residual_width, time_minutes = as.numeric(NPBboot.mll[[2]], units = "mins")))
```

```
## residual_width   time_minutes
##      0.9432024      3.0978285
```

Comparison

Having constructed all relevant confidence intervals, we can now compare the methodologies and their respective outputs. Before doing this it is important to note that we have constructed normal confidence intervals, this places an extra assumption on the calculation of the intervals, thus making them narrower. However, as we have standardised this approach across all the methodologies, the inference on the width of the intervals is valid for comparison.

In the analytic confidence interval, numerous restrictions and assumptions are made throughout. These include assymptotic assumptions, how unknown quantities are estimated and the use of pilot runs to select the bandwidth. This can have the effect of reducing the width of the confidence interval due to the constraints reducing your uncertainty about the estimation. This is shown clearly through the calculations of the width where the analytic width is shown to be smaller than the bootstrap methods.

In the bootstrap confidence intervals, the semi-parametric residual bootstrap assumes the model is correct up to a standard error, as shown through the use of residuals. In contract the nonparametric paired bootstrap makes no assumptions, this suggests that it should have the widest confidence interval due to it having the most uncertainty. This is again shown to be true as the paired width is wider than the residual width. It is worth noting here that the increased certainty has the benefit of a reduced computation time in comparison to the bootstrap methods, there is a payoff to be had for a better confidence interval and computation time must be taken into consideration when choosing your interval.

What we have shown through this comparison is that the more restrictions you impose, the more sure you are about your estimator and subsequently the narrower the confidence interval. This concludes our discussion of the confidence intervals and their respective widths. As for the specific application to the data, it is important to acknowledge the bias component for the analytic confidence interval. For smaller datasets you would have to apply undersmoothing to the optimal bandwidth as suggested in Hall (1992) in order to asymptotically remove this term. There is little comparison to be made here about the confidence intervals with respect to the data itself, other than the large sample of data has clearly reduced the width of the confidence intervals in comparison to the examples seen in lectures.

Conclusion

Throughout this report we have derived and constructed various confidence intervals for the local linear regression estimator and applied it to the 'openair' dataset. Within our investigation we have assessed and compared the confidence intervals and their widths. There is little that can be taken from the outputs above relative to our application within the specifications for the report. However, it is clear to see that the estimator is clearly a good fit and suggests that the mean concentration of pollutants is decreasing over time in the UK. For further research, it would be beneficial to look into the assumptions made by the intervals and potentially attempt the analytical confidence intervals through the use of different estimators.