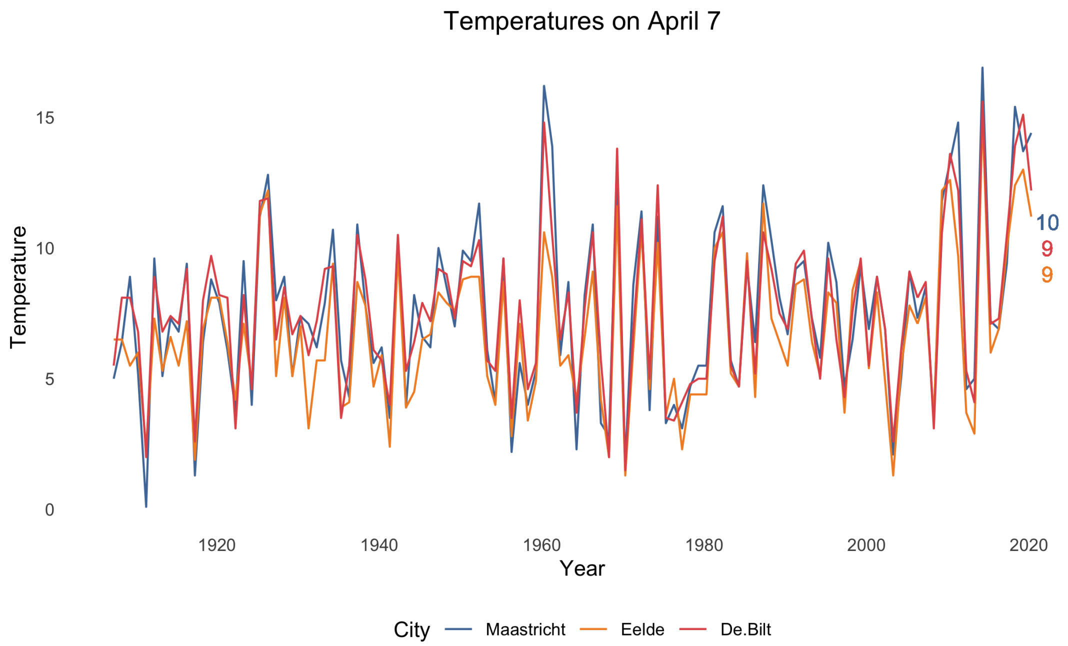
Project Mathematical Statistics
(EBC2107)

An Empirical Investigation of Temperatures in The Netherlands

Chumasha Rajapakshe (i6245577)
Tobias Schnabel (i6255807)

Tutorial Group 2
Robert Adámek



Contents

Part 1: Report

Introduction

Climate change is an important topic politically and socially, which translates to an equal importance scientifically. There are few topics in the public discourse which are characterized by the same heated debate about scientific facts and evidence needed to make policy as much as climate change, or, to call it by its original name, global warming (Magnus, Melenberg, and Muris 2011). Although there is a broad consensus in the scientific community, many people refuse to believe warnings by climate scientists, often citing supposed fabrication of evidence.

This report aims to answer one question: is there statistical evidence of an upward trend in temperatures in the Netherlands? We answer this question to the best of our ability using the tools and techniques covered in this course. We carefully discuss the assumptions we make, the methods we use, how robust those methods are, and why we believe that they are adequate.

Specifically, we provide statistical evidence to answer two individual questions:

1. Can we use statistical tests to determine whether average temperatures now are higher than they were last century?

2. Can we demonstrate the existence of an upward linear trend in temperatures over time?

We tackle these questions sequentially, preceded by a discussion of our data and assumptions, and followed by a respective discussion of the robustness of our results.

1.1 Descriptive Statistics

The data used in this paper are based on one panel data set that captures daily temperatures measured in three cities in the Netherlands: De Bilt, Eelde, and Maastricht. As these cities are not located in the same region of The Netherlands, their respective temperatures differ. On the basis of this panel data set, one can construct "collapsed" data sets that report means of temperatures in longer intervals. For the purposes of this paper, we confine ourselves to three levels of data granularity: daily, monthly, and yearly. As we will soon discover, the daily data is nor very well-suited for our purposes, as it is both very noisy (that is, it shows large variance and contains significant outliers) and bimodally distributed. We will therefore mainly use the monthly and yearly level panel data sets. Unless otherwise stated, most computations in the subsequent sections are performed for the entire data set for both of these levels, that is, for each city and each month (or year). In the interest of brevity, we will discuss our results for one city and point the interested reader to appendices A and B, where all results are shown for completeness. Throughout this paper and the appendices, we present confidence intervals along with our point estimates. We do so to quantify our estimation uncertainty, and we do not belabor the interpretation of these CIs as we trust the reader to be able to interpret them.

1.1.1 Entire Sample

Our data show temperatures in the range of $-17.4°C$ to $-30.9°C$, for a total of $41,639$ observations at the daily level.

Table 1.1: Daily Data

Statistic	N	Mean	St. Dev.	Min	Max
De Bilt	41,639	9.576	6.299	−14.900	29.700
Eelde	41,639	8.860	6.411	−17.400	28.300
Maastricht	41,639	9.705	6.665	−16.300	30.900

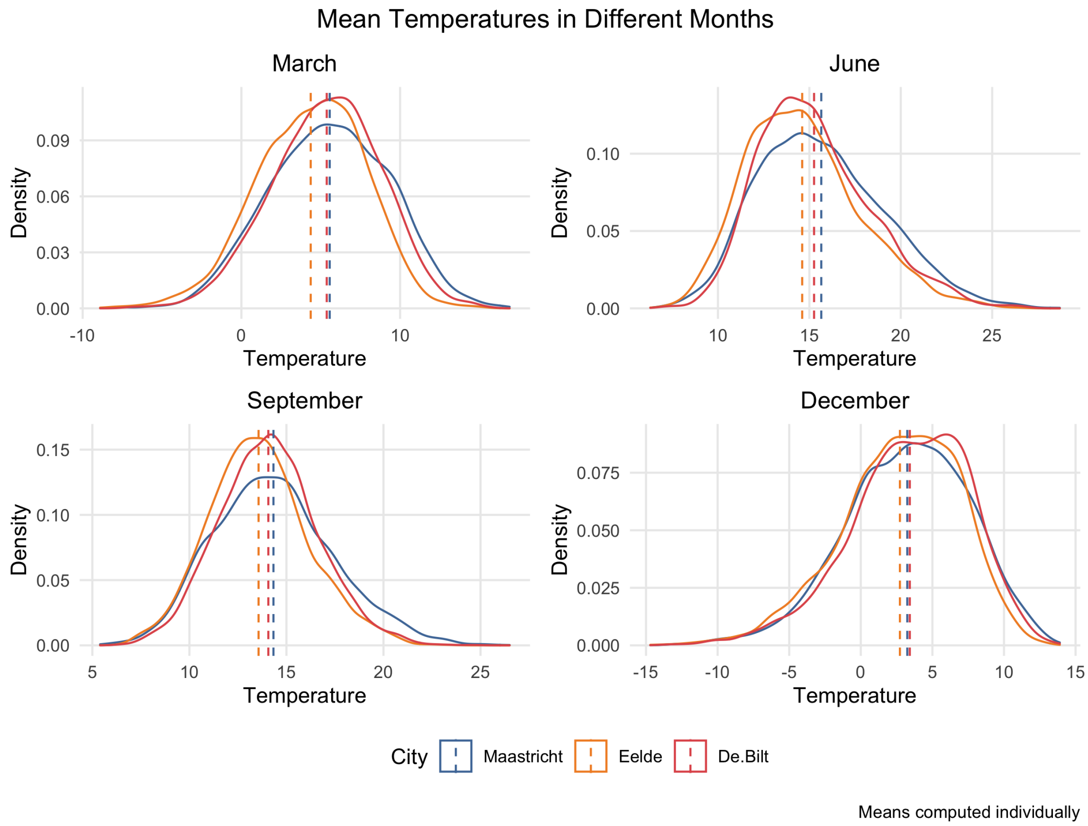
Figure 2.1 shows a density plot of this data. As we can see, it is bimodally distributed, which we attribute to seasonality. Figure 2.10 shows us that the quantiles of the distribution of this daily data are fairly close to that of a $\mathcal{N}(0, 1)$ distribution. Due to the clear presence of seasonality and the nosiness of this data, we only use it to generate subsets that are of interest to us.

The monthly data are less precise due to the lesser granularity of measurement, which is reflected by the smaller temperature range of $-7.6°C$ to $22.9°C$, but they present the same problems as the daily data: noisiness and bimodal distribution. Figures 2.2 and 2.3 show that their distribution is again bimodal, though this bimodality is more pronounced than before, which can also be seen in figure 2.11. Using custom functions we wrote, we can, however, subset this monthly data further to try to gain information on seasonality. Figure 1.2 below (2.9) shows the respective distributions of temperatures in the final months of each quarter: March, June, September, and December.

Table 1.2: Monthly Data

Statistic	N	Mean	St. Dev.	Min	Max
De Bilt	912	12.553	4.004	1.293	22.310
Eelde	912	11.912	4.065	0.343	21.387
Maastricht	912	12.781	4.245	0.593	22.913

Figure 1.1



Looking at the individual panels, we can see that the respective temperatures all appear to follow a Bell Curve, albeit they show skew in different directions and to varying extents. Unsurprisingly, the means of the distributions in December are located approximately 10°C below those in June, which explains the bimodality mentioned above. One possible way to address this seasonality is to "detrend" or "smooth" the monthly data using, for example, a linear smoother:

$$Y_i^s = \frac{1}{24}Y_{i-6} + \frac{1}{12}\sum_{j=-5}^{5} Y_{i+j} + \frac{1}{24}Y_{i+6} \tag{1.1}$$

where Y_i and Y_i^s denote the temperature and smoothed temperature in month i, respectively. While this process solves our problem of bimodality (which makes for example the construction of confidence intervals substantially harder, to name just one such problem), it creates a new one: as the Y_i^s are computed using surrounding values, 1.1 ensures that our smoothed monthly data are not independent, which would present major problems if we wanted to use them in subsequent estimations. Figures 2.4 and 2.12 show us that these smoothed monthly data are at first glance much "nicer" in terms of distribution, but due to the difficulties presented by their lack of independence, we will not use them or data smoothed in another way moving forward. Another way to adress this seasonality is to construct overlapping or "rolling subsamples", which we demonstrate in tables 2.8 and 2.9. The noisiness and sheer number of subsamples generated by this method are the reasons we do not use this method either. Rather, we make use of our subsetting functions to drop all observations of the the coldest months of the year, that is, December, January, February, and March, from the monthly data. Figures 2.2 and 2.3 show the difference in distribution that results from this choice. Section 2.6 motivates this choice further. **From this point onward, any reference to "monthly data" is referring to this subsetted data.** See Appendix D, lines 164 - 204

We now turn our attention to the least granular, that is, annual, level of data.

Figure 1.2

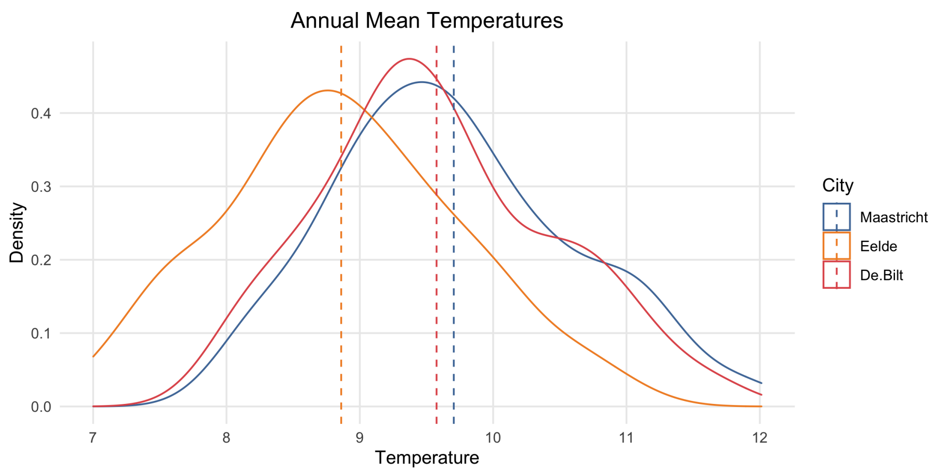
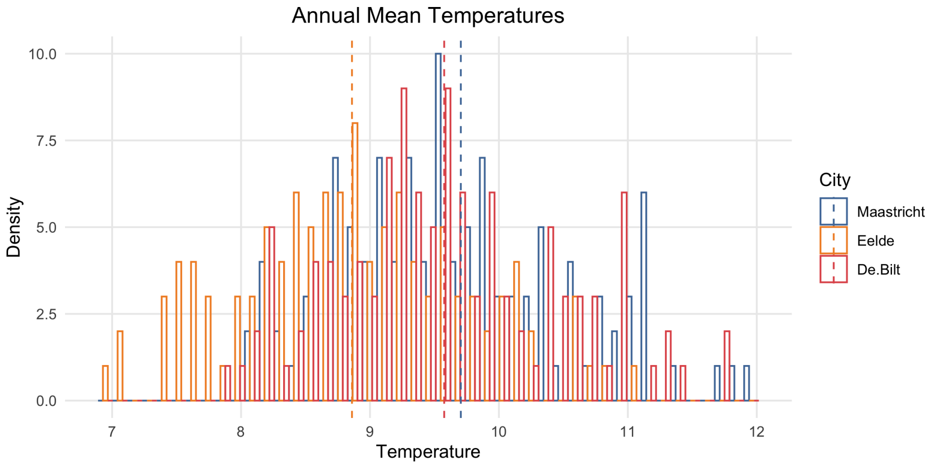


Figure 1.3



The annual data is again less precise than the monthly data (as is evident when comparing1 tables 2.1 - 2.4. It is also, however, distributed in a shape somewhat similar to that of a normal distribution. Figure 2.13 shows a Quantiles-to-Quantiles plot of the yearly data against a $\mathcal{N}(0, 1)$ distribution. As our data roughly follow a 45 degree line, we will assume that they are approximately normally distributed moving forward. We do so for multiple reasons: simplicity, inability to perform our inferential analysis otherwise, scope of this paper and the corresponding course, and an appeal to asymptotic arguments as outlined in section 2.

1.1.2 Subsamples

To aid our investigation of climate trends, we subset both our annual and our monthly data in two ways each. First, we run a Chow test to investigate the presence of a structural break (a point at which our times series data chagnes abruptly) in our data Chow (cf. 1960, section 5). Our highly statistically significant test results are indicative of the presence of a structural break in both the yearly and the monthly data. We then run an automated test for when this break point occurs, pursuant to Bai and Perron (2003). Our results indicate that this structural break occurs either at the end of 1961 or the beginning of 1962. It is worth mentioning that the authors at present do not understand either of these tests sufficiently to draw conclusions about their accuracy, applicability, or robustness. We therefore continue to include results for our estimations using this "empirically found" structural break, but focus our attention on a different structural break: one that occurs in roughly 1975.

This year is not chosen arbitrarily, but rather based on findings such as those in Zhou et al. (2009) and Sarkar and Maity (2021), which reference about a dozen papers each concerned with showing the existence and the timing of such a structural break in temperatures. As the authors of this paper are not experts in climatology by any stretch of the imagination, we are happy to take those findings at face value and hypothesize that a structural break in temperatures occurred in 1975.

1.2 Assumptions about our Data and Model

In order for us to perform inferential statistics on our data, we need to assume certain things about both the *data-generating process* (DGP) and the features of our sample data. We quickly discuss and motivate these assumptions in this section, although we would like to note that while there are standard tests we could do for most of these, we have chosen not to do so in some instances. We follow the structure of Wooldridge (2015, Ch. 10).

1.2.1 Linearity

We assume that the temperature in each city can be correctly modelled as a stochastic process $\{(x_{t1}, x_{t2}, \ldots, x_{tk}, y_t) : t = 1, 2, \ldots, n\}$ which follows a linear model

$$y_t = \beta_0 + \beta_1 x_{t1} + \ldots + \beta_k x_{tk} + u_t \tag{1.2}$$

where $\{u_t = 1, 2, \ldots, n\}$ is the sequence of errors (and n denotes the number of time periods or observations) (cf. Wooldridge 2015, TS.1).

1.2.2 Absence of Perfect Collinearity

We assume that "*in the sample (and therefore in the underlying time series process), no independent variable is constant nor a perfect linear combination of the others*" (cf. Wooldridge 2015, TS.2, p. 340).. As there is no reason to believe that years (or months) can be represented as a linear combination of other years (or months) across the sample, we do not test this assumption.

1.2.3 Zero Conditional Mean

We assume that $E(u_t|\mathbf{X}) = 0, t = 1, 2, \ldots, n$ (cf. Wooldridge 2015, TS.3). This assumption automatically holds if $u_t \perp\!\!\!\perp \mathbf{X}$ and $E(u_t) = 0$, which we assume. In other words, we are assuming strict exogeneity of our explanatory variables (which is probably a bit of a stretch, but necessary given the limited scope of the methods covered in this course).

1.2.4 Homoskedasticity

We assume that $Var(u_t|\mathbf{X}) = Var(u_t) = \sigma^2, t = 1, 2, \ldots, n$ (cf. Wooldridge 2015, TS.4), that is, we assume homoskedastic error terms. We test this assumption using a White test (cf. Wooldridge 2015, 12-6b), as this test can capture nonlinear forms of heteroskedasticity that the better-kown Breusch-Pagan test cannot (cf. Wooldridge 2015, 8-3a). This test has homoskedasticity as its Null Hypothesis, the results are reported in tables 2.39 and 2.40 and are not indicative of heteroskedasticity.

1.2.5 Absence of Serial Correlation, Independence

We assume that $Corr(u_t, u_s|\mathbf{X}) = 0 \; \forall t \neq s$ or, alternatively, $Corr(u_t, u_s) = 0 \; \forall t \neq s$, which simply means that the error terms of our linear model(s) are uncorrelated through time. This is automatically satisfied if our DGP satisfies independence (cf. Wooldridge 2015, TS.5).

1.2.6 Normality

Lastly, as mentioned in Section 1.1.1, we assume that our $u_t \perp\!\!\!\perp \mathbf{X}$ *and* $u_t \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2)$. This assumption is necessary in order for us to be able to use Ordinary Least Squares (OLS) standard errors, t- and $F-$statistics (cf. Wooldridge 2015, TS.6). Although there are methods to test for normality (cf. Lobato and Velasco 2004), those tests go beyond the scope of this course and we content ourselves with looking at 2.5, 2.6, 2.11, 2.13, and 2.20 to conclude that this assumption holds sufficiently (which again is a bit of a stretch but necessary for the feasibility of the rest of this paper).

1.2.7 Gauss-Markov and Normality of Sampling Distributions

The main benefits of assuming linearity, lack of perfect collinearity, zero conditional mean, homoskedasticity, and absence of serial correlation (i.e. Wooldridge 2015, TS.1-TS.5) is twofold: since we assume that they hold, we know that $\hat{\sigma}^2 = \frac{SSR}{df}$ is an unbiased estimator of σ^2 (cf. Wooldridge 2015, Theorem 10.3) and, more importantly, the OLS estimators used in Section 1.4 are the Best Linear Unbiased Estimators conditional on $\mathbf{X}$ (cf. Wooldridge 2015, Theorem 10.4 (Gauss-Markov)). Forthcoming research on the required strength of the assumptions in this theorem even suggests that the OLS estimators are *Best* Unbiased Estimators, not just Best *Linear* Unbiased Estimators (Hansen 2022, cf.)

The additional benefit of assuming Normality, which implies zero conditional mean, homoskedasticity, and lack of serial correlation, but is stronger than those because of the independence and normality assumptions, lies in Wooldridge (2015, Theorem 10.5 (Normal Sampling Distributions)): *"the OLS estimators are normally distributed, conditional on* $\mathbf{X}$*. Further, each t-statistic* $\sim t$*, each F-statistic* $\sim F$*, and the usual construction of Confidence Intervals is valid".* This result is essential for the inferential statistics and regressions in the following two sections. The assumption that $u_t \perp\!\!\!\perp \mathbf{X}$ cannot be reliably tested empirically, rather we have to make due with an argument about the nature of our DGP. Absent any expertise in Physics, we will use the favorite trick of economists everywhere and simply assume that it holds.

An alternative way of motivating this assumption would be to appeal to the asymptotic distributions of our sampling distributions by a combination of the Central Limit Theorem, the (Weak) Law of Large Numbers, Slutsky's Theorem and the Continuous Mapping Theorem following Casella and Berger (2002, section 5.5). Using this line of argument, we could motivate our inference and regression results to be valid without making distributional assumptions (such as normality) **on the data themselves**. Given the sizes of our samples and the nature of parts of our bootstrap results, we refrain from making this argument because we do not believe that it applies.

1.3 Inferential Statistics

1.3.1 Comparing Means

We begin our statistical inference by comparing the mean temperature of the three cities before and after the 1975 structural change. Specifically, we compute a *paired t-test* as shown in Casella and Berger (2002, ex. 8.39). We have $(X_1, Y_1), \ldots, (X_n, Y_n)$ where $X_i \overset{i.i.d}{\sim} \mathcal{N}(\mu_X, \sigma_X^2)$ and $Y_i \overset{i.i.d}{\sim} \mathcal{N}(\mu_Y, \sigma_Y^2)$. In order to be able to test $H_0 : \mu_X = \mu_Y$ versus $H_1 : \mu_X \neq \mu_Y$, we first compute $W_i := X_i - Y_i$, which are $W_i \overset{i.i.d}{\sim} \mathcal{N}(\mu_W, \sigma_W^2)$, and then our test statistic

$$T_W = \frac{\overline{W}}{\sqrt{\frac{1}{n} S_W^2}} \sim t_{n-1} \tag{1.3}$$

where $\overline{W} = \frac{1}{n} \sum_{i=1}^{n} W_i$ and $S_W^2 = \frac{1}{n-1} \sum_{i=1}^{n} (W_i - \overline{W})^2$. Table 1.3.1 shows the results of these calculations, as well as the corresponding p-values and Confidence Intervals.

Table 1.3: t-tests, 1975 Break

	t-Statistic	p-value	C.I. Lower	C.I. Upper
De Bilt	7.5057	0.0000000	0.7923	Inf
Eelde	5.7760	0.0000004	0.6173	Inf
Maastricht	7.0249	0.0000000	0.7513	Inf

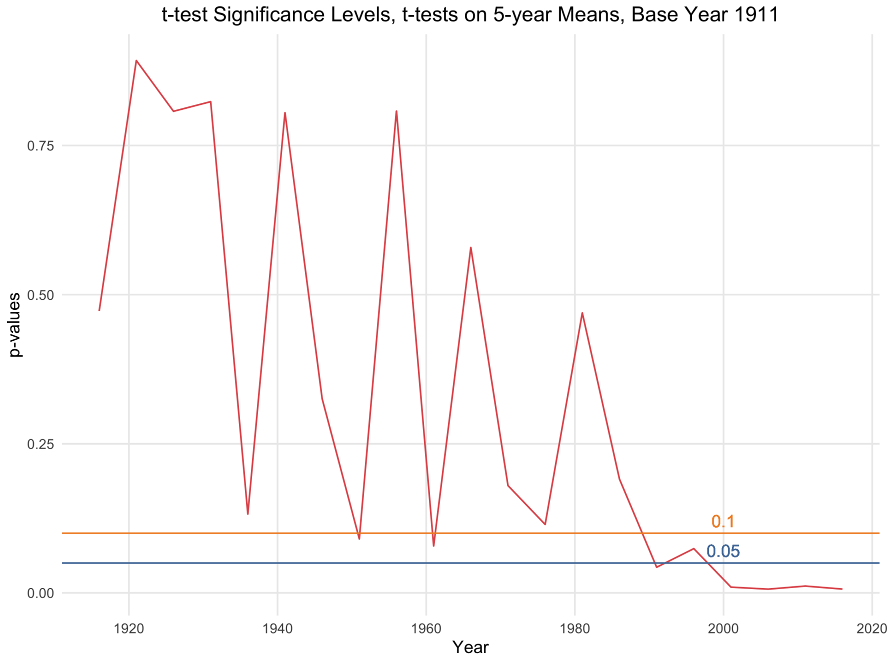
Applied to our data, this yields $W_i := X_{city\,i,after1975} - X_{cityi,before1975}$ and $H_0 : \mu_W \leq 0$ versus $H_1 : \mu_W > 0$ (right-tailed test). As table 1.3.1 shows, the tests for all three cities are highly statistically significant, that is, we reject the Null Hypothesis that the mean Temperature of each city is not greater before 1975 than it is after 1975.

We repeat these tests for both the yearly and the monthly data for the 1975 and 1961 breaks, the results are reported in tables 2.12 to 2.15 and mirror the one just obtained.

We then repeat this exercise for a subset of our data that aggregates the mean and median (for variety's sake) temperatures over non-overlapping intervals of 10 years each using custom functions, the results of which are shown in tables 2.6 and 2.7 ("Base" indicates the year which we take to compute μ_0, the tests statistics are negative because we subtract the later group's mean from the earlier one's). Results are similar, but only tests where the difference in years between the groups is at least 80 are statistically significant.

Lastly, we repeat this exercise yet again for a subset of five-year means (see table 2.8, same caveats). For this last variation, we plot the p-values associated with these tests:

Figure 1.4



In this narrower time window, only the tests for the windows beginning in 1991 (tested against a base year of 1911) are statistically significant. We do not take this to mean that these tests are better than tests for windows before 1991 (or that the latter are less valid), but it does show that when using these relatively narrow five-year intervals, noticeable differences only start to appear after longer time differences (one reason for the low power of earlier test probably lies in the small sample size used to compute the test-statistic as a result of only aggregating five years at a time).

1.3.2 Comparing Variances

The tests performed in the previous section ordinarily rely on an assumption of equal variances (though we circumvented this issue by using R's built-in functionality of separate variance estimation for both groups and using the Welch modification to the degrees of freedom (cf. e.g. Casella and Berger 2002, 5.3.5ff.)). We now formally test whether this is indeed the case using an F-test (Snedecor and Cochran 1967, cf. e.g.), which uses $(X_1, \dots, X_n)$, $(Y_1, \dots, Y_n)$, where $X_i \overset{i.i.d}{\sim} \mathcal{N}(\mu_X, \sigma_X^2)$ and $Y_i \overset{i.i.d}{\sim} \mathcal{N}(\mu_Y, \sigma_Y^2)$ with $H_0 : \sigma_X^2 =_Y^2$ versus $H_1 : \sigma_X^2 \neq \sigma_Y^2$ with a test statistic under H_0 of

$$F = \frac{S_X^2}{S_Y^2} \sim F_{N_X - 1, N_Y - 1} \tag{1.4}$$

and rejects if $F \notin [F_{1-\frac{\alpha}{2}, N_X-1, N_Y-1}, F_{\frac{\alpha}{2}, N_X-1, N_Y-1}]$.

Table 1.4: F-tests, Yearly Data, 1975 Break

	F-Statistic	p-value	C.I. Lower	C.I. Upper	Point Est.
De Bilt	0.462	0.0118367	0.2538	0.8405	0.4619
Eelde	0.509	0.0272475	0.2796	0.9258	0.5088
Maastricht	0.440	0.0075549	0.2418	0.8006	0.4400

Table 1.4 shows the results of this test for the yearly data assuming a 1975 structural break. The test is highly significant across all cities, hence we can reject the Null Hypothesis of equal variances of the sample both before and after 1975.

We repeat these tests for both yearly and monthly data for both the 1975 and 1961 breaks. The results are reported in tables 2.17 to 2.21. They show statistical significance for the annual, but not the monthly data each time, which we believe to be attributable to the loss in precision (and therefore also variance) when aggregating the monthly into the annual data.

1.3.3 Preliminary Result

By now, we have gathered enough evidence to answer the first of our two specific research questions: we can use specific statistical tests to determine whether average temperatures now are higher than they were last century. Our results resoundingly indicate that our data satisfy the assumptions we make in section 1.2. Further, our tests provide strong evidence of higher average temperatures over time, specifically shown in tables 2.12 to 2.15. Our results of unequal variances across subsamples is only milady problematic, as it does not affect our underlying assumptions or our results and can be mitigated using computational methods.

1.4 Regression Analysis

To answer our second question, we now turn our attention to the main model we use in accordance with our assumption of Linearity. We specify our model following Cunningham (2021), beginning with the population-level linear model:

$$Y_t = \beta_0 + \beta_1 X_t + u_t \text{ with } E(u_t|\mathbf{X}) = E(u_t) = 0 \qquad u_t \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma_Y^2)$$

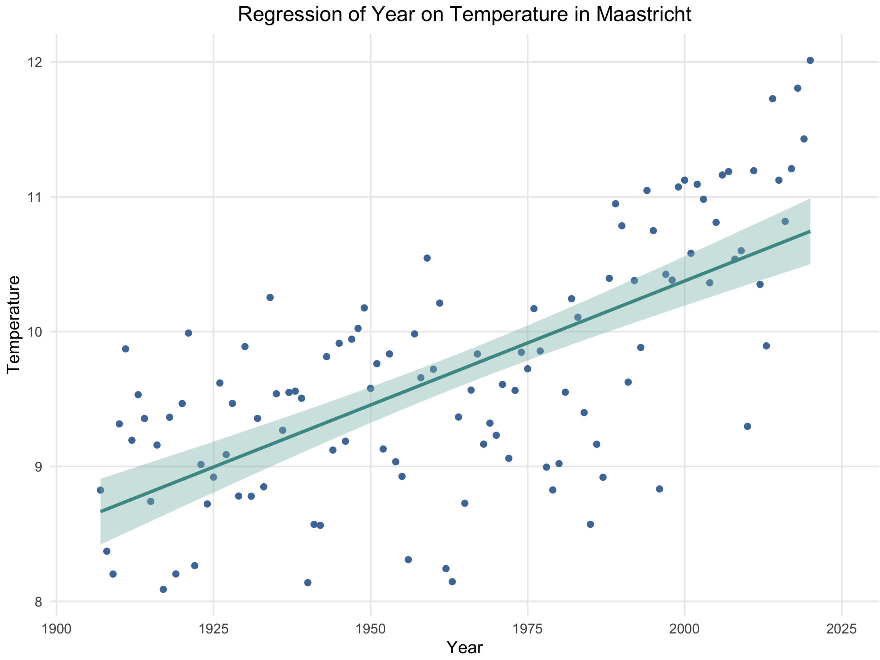
$$Y_i \overset{i.i.d}{\sim} \mathcal{N}(\beta_0 + \beta_1 \mathbf{X}, \sigma^2) \Rightarrow (Y_1, \dots, Y_n) \perp\!\!\!\perp \forall i = 1, \dots, n \qquad \widehat{\beta_0} = \bar{y} - \widehat{\beta_1}\bar{x}$$

$$\widehat{\beta_1} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})} = \frac{S_{xx}}{S_X^2}$$

$$\text{which leads us to our sample model} \qquad \widehat{Y} = \widehat{\beta_0} + \widehat{\beta_1}\mathbf{X}$$

To estimate the coefficient of this sample model, we program a function that takes as input the vectors $\mathbf{X}$ and y and returns point estimates, standard errors, and p-values (code using matrix notation can be found on page 69 of Appendix C) using the following main computations: given

Figure 1.5



$y_{n\times 1}$, $X_{n\times k}$, $u_{n\times 1}$, $\beta_{k\times 1}$, we have (cf. Wooldridge 2015, Advanced Treatment E)

$$Y = X\beta + u \text{ (population-level model)} \qquad e = y - X\beta \text{(residuals)} \tag{1.5}$$

$$RSS = e'e \text{ (Sum of squared Residuals)} \qquad RSS = y'y - 2\hat{\beta}'X'y + \hat{\beta}'X'X\hat{\beta} \tag{1.6}$$

$$\text{to find our } \hat{\beta}: \qquad \frac{\partial e'e}{\partial \hat{\beta}} = -2X'y + 2X'X\hat{\beta} \overset{!}{=} 0 \tag{1.7}$$

$$\iff (X'X)\hat{\beta} = X'y \qquad \hat{\beta} = (X'X)^{-1}X'y \tag{1.8}$$

$$\hat{y} = X\hat{\beta} \text{ (sample-level regression line)} \qquad \hat{u} = y - \hat{y} = y - X\hat{\beta} \tag{1.9}$$

$$Var(\hat{\beta}|X) = \sigma^2(X'X)^{-1} \text{ (Variance-Covariance Matrix)} \tag{1.10}$$

$$SER = E(\sigma^2) \overset{GM}{=} \sigma^2 \text{ (SE Regression)} \qquad GM = \text{Gauss-Markov} \tag{1.11}$$

1.4.1 Entire Sample

Tables 1.5, 2.22 to 2.27 show results for these manual computations. The respective city's temperature serve as the dependent variables, whereas year (for the annual data) and month (for the monthly data) serve as the independent variables.

Table 1.5: Manually Computed Regression Coefficients, Maastricht, Yearly Data

	estimate	se	p-value
alpha	-26.427605	3.693529	0.00000000
beta	0.018402	0.001881	0.00000000

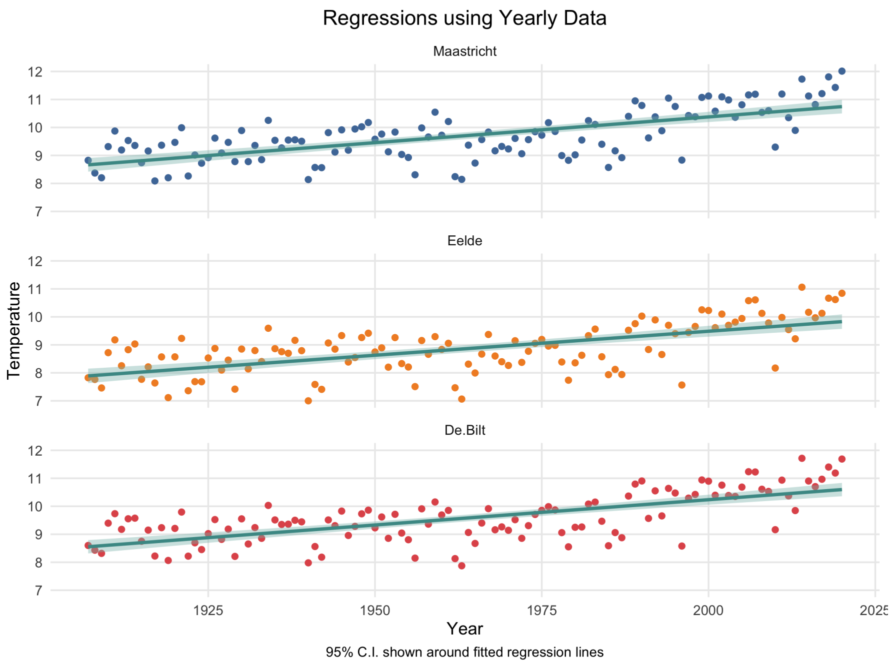
To avoid confusion while programming, we also denote β_0 as α, so that the sample-level regression equation for Maastricht using Yearly data would be

$$\begin{aligned}\widehat{y_{Maas}} &= \widehat{\beta_0} + \widehat{\beta_1} x \\ \widehat{y_{Maas}} &= -26.42 + 0.018402 * year\end{aligned} \tag{1.12}$$

The interpretation of these results is as follows: for the slope coefficient $b\hat{e}ta_1(\hat{\beta})$ $\Delta\hat{y} = \hat{\beta}_1 \Delta x$ denotes the *partial effect* of x, which is to say: ceteris paribus, temperatures in Maastricht increase by $0.0184°C$ per one-unit (i.e. 1-year) increase in $year$. The intercept $\hat{\beta}_0(\hat{\alpha})$ is the predicted value of y (i.e. temperature) when $year = 0$. As in many econometric applications, the intercept in our case does not have a meaningful interpretation (Wooldridge 2015, pp. 3–2).

The results shown in tables 2.22 to 2.27 all show estimated slope coefficients of positive sign with a value of about $0.018°C$ for the yearly and $0.00018°C$ for the monthly data. Figure 1.6 below shows visualizations of estimated specifications of our model (eq. 1.9).

Figure 1.6: ARegsY



Upon obtaining these results, we lastly have to check whether they are statistically significant. We follow the standard[1] approach for testing the statistica significance of our coefficients, which is to set $H_0 : \hat{\beta}_1 \leq 0$ versus $H_1 : \hat{\beta}_1 > 0$. We compute the p-values, that is, the probability of observing a value for $\hat{\beta}$ and α at least as extreme as those computed given that the Null Hypothesis is true, i.e. given that $\hat{\beta}_1 \leq 0$, as we would compute those in a t-test, exploiting the fact that $\frac{\hat{\beta}}{se} \sim t_{n-1}$, where se denotes the standard error, i.e. the square root of the main diagonal of the Variance-Covariance Matrix (eq. 1.10). All results are statistically significant at the 1%

[1] in this case, we specifically use a right-tailed test rather than the more common two-tailed one, which is due to the specification of our model and our knowledge that β is positive on the population level

confidence level, which we can also see in the (better presented) tables 2.33 and 2.34. Those latter tables also include information about the estimated R^2 (goodness of fit) and F-statistic (statistical significance of the entire model). We include these tables mainly for the interested reader (i.e. grader), but will not belabor the interpretation of regression statistics that are clearly outside the scope of this course.

1.4.2 Restricted Sample

As in our discussion of inferential statistics, we now extend our linear model to include a structural break. As before, we focus on a hypothesized break in 195 and direct the interest reader to the appendix for results about our "empirical" break in 1965.

Figure 1.7

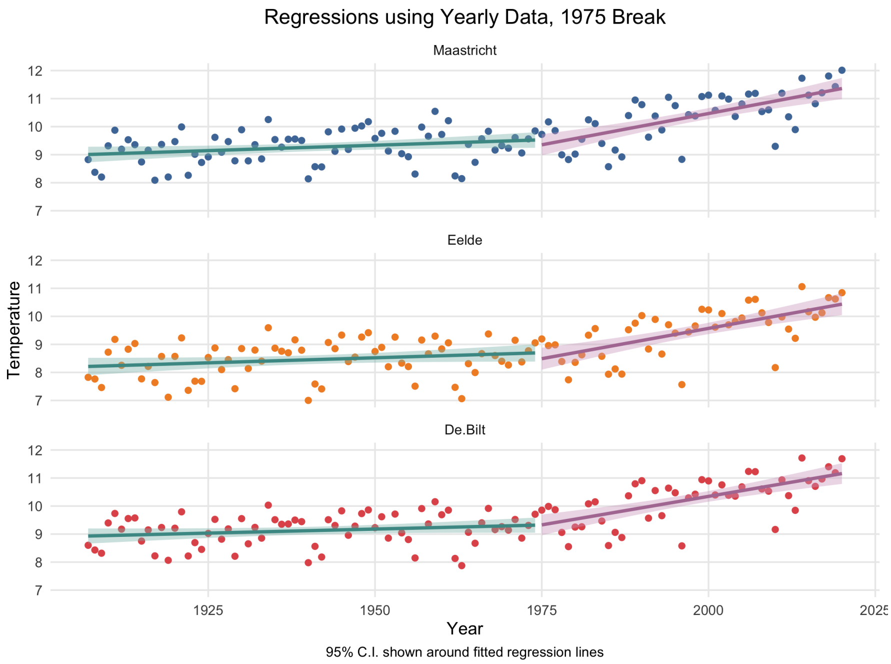


Figure 1.7 shows a visualization of our regression results. For lack of econometric sophistication, we simply apply our model (1.5) to two separate time periods: one before and one after 1975. As the interpretation of our estimated coefficients is the same, we can directly look at tables 2.30 - 2.37 and 1.6 below to see that the estimated slope coefficients are higher after the 1975 break. In all cases and in both the yearly and monthly data, the estimated slope coefficients are not just higher after the 1975 (and 1961) break point, but are also statistically significant at at least the 5% significance level, which most estimated slope coefficients for the period before the break point are not.

Additionally, as the bottom columns of tables 1.6 and 2.30 - 2.37 show, the estimated regression lines after the break points have dramatically better fit and joint significance than those before the 1975 (and 1961) break points. Our earlier comment notwithstanding, we note that these measure of fit lead us to believe that the apparent upward linear trend in temperatures we find accelerated significantly around the 1970s, which is very much in line with the literature cited in Sarkar and Maity (2021) and Zhou et al. (2009)

Table 1.6: Regressions, Yearly Data, Before and After 1975 Break

	Dependent variable:	
	maastricht	maastricht
	Year < 1975	Year > 1975
	(1)	(2)
year	0.003	
	(0.007)	
year		0.046***
		(0.007)
Constant	3.046	−81.052***
	(13.182)	(14.872)
Observations	45	45
R^2	0.005	0.468
Adjusted R^2	−0.018	0.455
Residual Std. Error (df = 43)	0.588	0.648
F Statistic (df = 1; 43)	0.231	37.789***
Note:	*p<0.1; **p<0.05; ***p<0.01	

1.4.3 Preliminary Results and Robustness Checks

We have now gathered sufficient evidence to answer our second research question: we can indeed demonstrate the existence of an upward linear trend in temperatures over time. Our results unequivocally show that there exists a linear relationship between temperatures and time which has a statistically significant positive slope. This is, however, also the time to caution about over-interpreting the (non-)statistical significance of our results. Simply running linear regressions is by no means a causal research design, the many reasons for which are discussed in a better way than we could dream to achieve by authors such as Cunningham (2021) and Huntington-Klein (2021). It is plausible, if not perhaps probable, that our crude statistical methods are subject to problems such as heteroskedasticity, serial correlation, lack of independence, and nowhere-nearperfect normality, which we conveniently assumed away, though we are confident that the overall sign and significance of our findings is correct.

1.5 Bootstrap Analysis

One way of investigating the robustness of our results is to *bootstrap* our sample , that is, to re-sample with replacement from our sample, perform inference on those re-sampled samples, and aggregate this information to draw inference on the population. As the motivation and derivation of this method are discussed in depth by a.o. Casella and Berger (2002, 478 ff.), we will turn directly to the application, which we have split into three parts. The accompanying code can be found in Appendix D.

1.5.1 Paired t-test

We use our previously obtained subsamples to run a bootstrapped (10,000 iterations) paired t-test with $H_0 : \mu_{year<1975} \leq \mu_{year>1975}$ versus $H_1 : \mu_{year<1975} > \mu_{year>1975}$. The respective test statistics for both the yearly and the monthly data as well as the accompanying p-values are shown in 1.7 below. All tests are highly statistically significant, which supports our earlier conclusion. We note that the confidence intervals shown in 1.5.1 are consistently slightly narrower than those found earlier.

Table 1.7: Bootstrap: Paired t-test

	t_n	p-value	CI lower	CI upper
De Bilt, Yearly Data	7.5057	0.000000	0.742881	1.284299
Eelde, Yearly Data	5.7760	0.000000	0.564044	1.167805
Maastricht, Yearly Data	7.0249	0.000000	0.699989	1.265793
De Bilt, Monthly Data	9.3368	0.000000	0.702866	1.075991
Eelde, Monthly Data	7.1676	0.000000	0.504343	0.888277
Maastricht, Monthly Data	8.3249	0.000000	0.681752	1.105389

1.5.2 Bootstrap on the Linear Model: Pairs and Residual Bootstrap

Lastly, we implement two separate bootstrap methods to investigate the robustness of our regression results. Using our previous linear model, which we do not need to assume to be true due to the Best Linear Predictor properties of the Least Squares estimator, we now drop all parametric distributional assumptions and instead assume that $(X_1, Y_1), \ldots, (X_n, Y_n)$ are a random sample with bivariate CDF F (estimated using the Empirical Distribution Function $\hat{F}$). We consider the asymptotically pivotal quantity

$$Q(\mathbf{Y}, \mathbf{X}, F) = Q_n'(\mathbf{Y}, \mathbf{X}, \beta) = \frac{\hat{\beta}_{n,LS} - \beta}{\sqrt{S_n^2 / S_{XX}}}$$

where $S_n^2 = \frac{1}{n-2} \sum_{i=1}^n (Y_i - \hat{\beta}_{0,LS} - \hat{\beta}_{n,LS} X_i)^2$ and its bootstrap version

$$Q^*(\mathbf{Y}^*, \mathbf{X}^*, \hat{F}_n) = Q_n'(\mathbf{Y}^*, \mathbf{X}^*, \beta^*) = \frac{\hat{\beta}^*_{n,LS} - \beta^*}{\sqrt{S_n^{*2} / S^*_{XX}}} \tag{1.13}$$

where $\hat{\beta}^*_{n,LS} = \frac{\sum_{i=1}^n (X_i^* - \overline{X}_n^*)(Y_i^* - \overline{Z}_n^*)}{\sum_{i=1}^n (X_i^* - \overline{X}_n^*)^2} = \frac{S^*_{XX}}{S_{XX}}$, it can be shown that for the pairs bootstrap where $\hat{F}_n = \hat{F}_n^E$, $\beta^* = \hat{\beta}_{n,LS}$. Defining c^*_α *s.t.* $P^*(Q_n^* \geq c^*_\alpha) = \alpha$, we can then construct the equal-tailed percentile-t interval for β:

$$\left[\hat{\beta}_{n,LS} - c^*_{\alpha/2} \sqrt{\frac{S_n^2}{\sum_{i=1}^n (X_i^* - \overline{X}_n^*)^2}},\ \hat{\beta}_{n,LS} - c^*_{1-\alpha/2} \sqrt{\frac{S_n^2}{\sum_{i=1}^n (X_i^* - \overline{X}_n^*)^2}}\right] \tag{1.14}$$

For the residual bootstrap, which is a less general method than the pairs bootstrap that works better for fixed X and benefits from imposing our knowledge of an linear model, on the other hand, we first calculate the residuals $\hat{u}_i = Y_i - \hat{\beta}_{0,n} - \hat{\beta}_{1,n} X_i$ of our linear model and apply the iid bootstrap to obtain the bootstrap errors $u_1^*, \ldots, u_n^*$, construct bootstrap regressors $X_1^*, \ldots, X_n^*$, build the bootstrap sample $Y_i^* = \beta_0^* + \beta_1^* X_i^* + u_i^*$ with $\beta_0^* = \beta^*_{0,n}$ and $\beta_1^* = \beta^*_{1,n}$ and repeat the steps of the pairs bootstrap starting in 1.13 to obtain 1.14. Table 1.8 below shows 1.13 as well as 1.14 for both methods and yearly as well as monthly data.

For $H_0 : \hat{\beta}_1 \leq 0$ versus $H_1 : \hat{\beta}_1 > 0$, we fail to reject the Null hypothesis in all cases. We conclude that we must have made assumptions and/or specification choices about our data and/or model that lead to significantly different results than those obtained using the less restrictive bootstrap methods. We suspect that the two assumptions most likely to be violated are that of normality and that of Zero Conditional Mean, a.k.a independence. This latter suspicion is due to the fairly well-established climatological reality that a global rise in temperatures has caused a rise in extreme weather events and drastic temperature swings (cf. e.g. Sarkar and Maity 2021). These are the reasons why we do not motivate our approach using the asymptotic properties of our estimators and tests. After all, if core assumptions of the model are plausibly invalid, there is no benefit in appealing to asymptotically nice behaviour of our methods.

Table 1.8: Bootstrap: t-test for Regression Coefficients

	Q^*	CI lower	CI upper
De Bilt, Yearly Data, Pairs	0.1369	-0.356699	0.397091
De Bilt, Yearly Data, Residuals		-0.712420	0.736694
Eelde, Yearly Data, Pairs	0.1405	-0.295937	0.339347
Eelde, Yearly Data, Residuals		-0.540706	0.731274
Maastricht, Yearly Data, Pairs	0.1357	-0.356233	0.401201
Maastricht, Yearly Data, Residuals		-0.704909	0.731274
De Bilt, Monthly Data, Pairs	0.1369	-0.107326	0.202439
De Bilt, Monthly Data, Residuals		-0.057493	0.058101
Eelde, Monthly Data, Pairs	0.1405	-0.139981	0.213268
Eelde, Monthly Data, Residuals		-0.066346	0.055207
Maastricht, Monthly Data, Pairs	0.1357	-0.110383	0.197385
Maastricht, Monthly Data, Residuals		-0.057359	0.055207

1.6 Conclusion

We set out two answer two research questions:

1. Can we use statistical tests to determine whether average temperatures now are higher than they were last century?
2. Can we demonstrate the existence of an upward linear trend in temperatures over time?

Sections 3 and 4 give a deceivingly clear answer to both, in that they both use objective methods such as t-tests and Least Squares estimation to answer both questions with a resounding "yes".

As the previous section on the bootstrap reveals, however, some of these conclusions, specifically those about the existence and strength of a linear trend, may have been a bit premature. The most crucial section in this paper is that containing our assumptions. We have good reason to believe that quite a few of the assumptions we make and discuss hold maybe approximately (or asymptotically), but do not hold for our yearly or monthly data.

One avenue for continued investigation of our research questions would be the extension of our methods (including the bootstrap methods) to correctly de-seasonalized monthly and daily data. Another might be the use of autoregressive models and of measures to adress serial correlation. Both of these avenues are out of reach for the authors of this paper due to a lack of statistical-econometric sophistication (and computational resources).

In closing, we believe that overall, our answer to our first research question is entirely valid, as backed up by our bootstrap results. We also believe that general sign and significance of our regression results to be plausible, though it appears that we either implemented the bootstrap methods incorrectly, misspecified our model(s), (and) or assumed too many real-world problems away. In any case, as the saying goes, our model might be wrong, but to us, it is nonetheless useful.

1.7 Software used

We used R version 4.1.2 R Core Team 2022a and the following R packages: evaluate v. 0.15 Wickham and Xie 2022, glue v. 1.6.2 Hester and Bryan 2022, grateful v. 0.1.11 Rodríguez-Sánchez, Jackson, and Hutchins 2022, highr v. 0.9 Xie and Qiu 2021, knitr v. 1.37 Xie 2022a, stringi v. 1.7.6 Gagolewski 2021, tidyverse v. 1.3.1 Wickham, Averick, et al. 2019, xfun v. 0.30 Xie 2022c, yaml v. 2.3.5 Garbett et al. 2022, zoo v. 1.8.9 **zoo**, running in RStudio v. 2022.2.1.461 RStudio Team 2022

References

Chow, Gregory C (1960). "Tests of Equality Between Sets of Coefficients in Two Linear Regressions". In: 28.3, pp. 591–605.

Snedecor, George W. (George Waddel) and William G. (William Gemmell) Cochran (1967). *Statistical methods*. 6th ed. Ames Iowa: Iowa State University Press, p. 593. ISBN: 9780813815602.

Pruscha, Helmut (1986). "A Note on Time Series Analysis of Yearly Temperature Data". In: *Source: Journal of the Royal Statistical Society. Series A (General)* 149.2, pp. 174–185. URL: https://www.jstor.org/stable/2981529.

Casella, George. and Roger L. Berger (2002). "Statistical inference". In: p. 660.

Bai, Jushan and Pierre Perron (Jan. 2003). "Computation and analysis of multiple structural change models". In: *Journal of Applied Econometrics* 18.1, pp. 1–22. ISSN: 1099-1255. DOI: 10.1002/JAE.659. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/jae.659%20https://onlinelibrary.wiley.com/doi/abs/10.1002/jae.659%20https://onlinelibrary.wiley.com/doi/10.1002/jae.659.

Lobato, Ignacio N and Carlos Velasco (2004). "A Simple Test of Normality for Time Series". In: 20.4, pp. 671–689. DOI: 10.1017/S0266466604204030. URL: https://www.jstor.org/stable/3533541.

Zhou, Tianjun et al. (Apr. 2009). "Why the Western Pacific Subtropical High Has Extended Westward since the Late 1970s". In: *Journal of Climate* 22.8, pp. 2199–2215. ISSN: 0894-8755. DOI: 10.1175/2008JCLI2527.1. URL: https://journals.ametsoc.org/view/journals/clim/22/8/2008jcli2527.1.xml.

Magnus, Jan R, Bertrand Melenberg, and Chris Muris (2011). "Global Warming and Local Dimming: The Statistical Evidence". In: *Source: Journal of the American Statistical Association* 106.494, pp. 452–464.

Wooldridge, Jeffrey M (2015). *Introductory econometrics: A modern approach*. Cengage learning. ISBN: 1305446380.

Cunningham, Scott (2021). *Causal Inference*. Yale University Press. ISBN: 0300255888.

Huntington-Klein, Nick (2021). "The effect: An introduction to research design and causality". In: ISSN: 1003226051.

Sarkar, Subharthi and Rajib Maity (2021). "Global climate shift in 1970s causes a significant worldwide increase in precipitation extremes". In: *Scientific Reports |* 11.123, p. 11574. DOI: 10.1038/s41598-021-90854-8. URL: https://doi.org/10.1038/s41598-021-90854-8.

Hansen, Bruce E (2022). "A MODERN GAUSS-MARKOV THEOREM". In: *Econometrica* forthcoming.forthcoming, pp. 1–11. DOI: 10.3982/ECTA19255. URL: https://doi.org/10.3982/ECTA19255.

Software Packages

Wickham, Hadley, Mara Averick, et al. (2019). *Welcome to the tidyverse.* 43, p. 1686. DOI: 10.21105/joss.01686.

Dowle, Matt and Arun Srinivasan (2021). *data.table: Extension of 'data.frame'.* R package version 1.14.2. URL: https://CRAN.R-project.org/package=data.table.

Gagolewski, Marek (2021). *stringi: Fast and portable character string processing in R.* R package version 1.7.6. URL: https://stringi.gagolewski.com/.

Wickham, Hadley, Winston Chang, et al. (2021). *ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics.* R package version 3.3.5. URL: https://CRAN.R-project.org/package=ggplot2.

Xie, Yihui and Yixuan Qiu (2021). *highr: Syntax Highlighting for R Source Code.* R package version 0.9. URL: https://CRAN.R-project.org/package=highr.

Garbett, Shawn P et al. (2022). *yaml: Methods to Convert R Data to YAML and Back.* R package version 2.3.5. URL: https://CRAN.R-project.org/package=yaml.

Hester, Jim and Jennifer Bryan (2022). *glue: Interpreted String Literals.* R package version 1.6.2. URL: https://CRAN.R-project.org/package=glue.

Hlavac, Marek (2022). *stargazer: Well-Formatted Regression and Summary Statistics Tables.* R package version 5.2.3. URL: https://CRAN.R-project.org/package=stargazer.

R Core Team (2022a). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/.

— (2022b). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/.

Rodríguez-Sánchez, Francisco, Connor P. Jackson, and Shaurita D. Hutchins (2022). *grateful: Facilitate citation of R packages.* R package version 0.1.5. URL: https://github.com/Pakillo/grateful.

RStudio Team (2022). *RStudio: Integrated Development Environment for R.* RStudio, PBC. Boston, MA. URL: http://www.rstudio.com/.

Wickham, Hadley, Romain François, et al. (2022). *dplyr: A Grammar of Data Manipulation.* R package version 1.0.8. URL: https://CRAN.R-project.org/package=dplyr.

Wickham, Hadley and Yihui Xie (2022). *evaluate: Parsing and Evaluation Tools that Provide More Details than the Default.* R package version 0.15. URL: https://CRAN.R-project.org/package=evaluate.

Xie, Yihui (2022a). *knitr: A General-Purpose Package for Dynamic Report Generation in R.* R package version 1.38. URL: https://yihui.org/knitr/.

— (2022b). *knitr: A General-Purpose Package for Dynamic Report Generation in R.* R package version 1.38. URL: https://yihui.org/knitr/.

Xie, Yihui (2022c). *xfun: Supporting Functions for Packages Maintained by 'Yihui Xie'*. R package version 0.30. URL: https://CRAN.R-project.org/package=xfun.

Part 2: Appendices

2.1 Appendix A: Figures

Figure 2.1: back to section 1.1.1

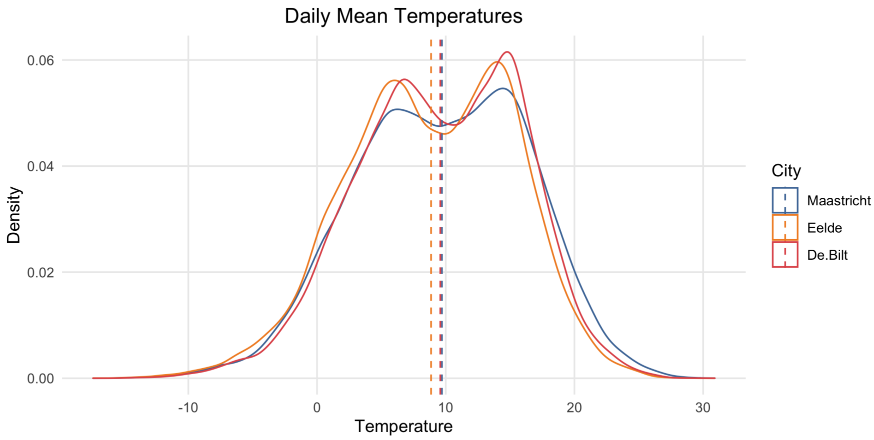


Figure 2.2: back to section 1.1.1

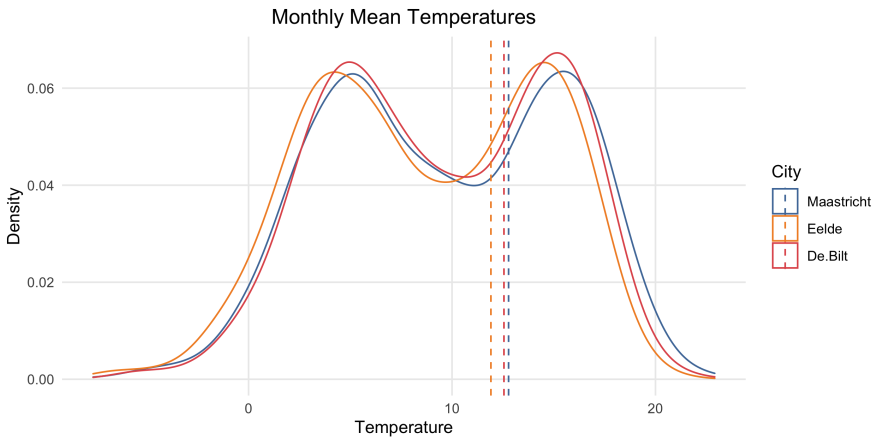


Figure 2.3: back to section 1.1.1

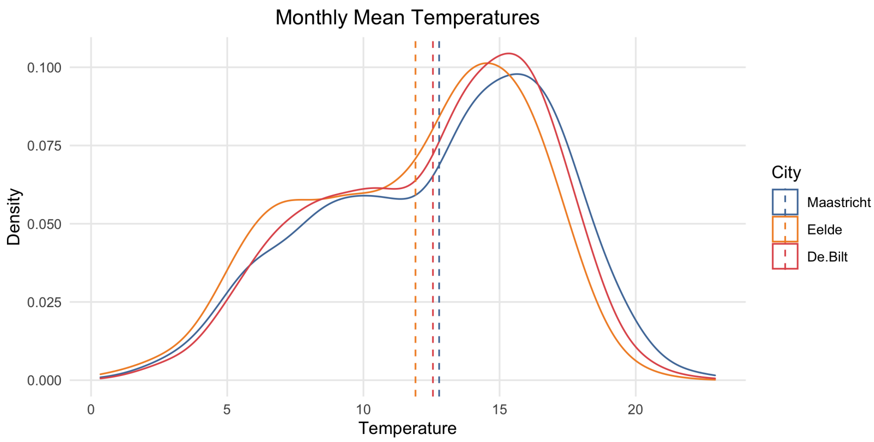


Figure 2.4: back to section 1.1.1

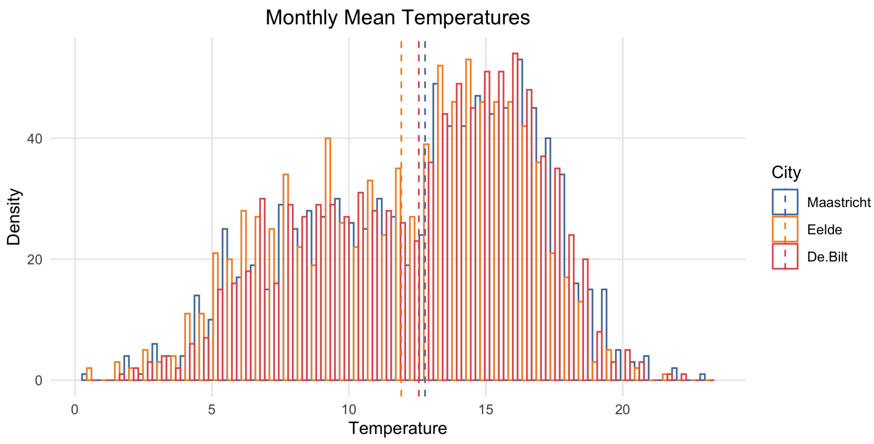


Figure 2.5: back to section 1.1.1

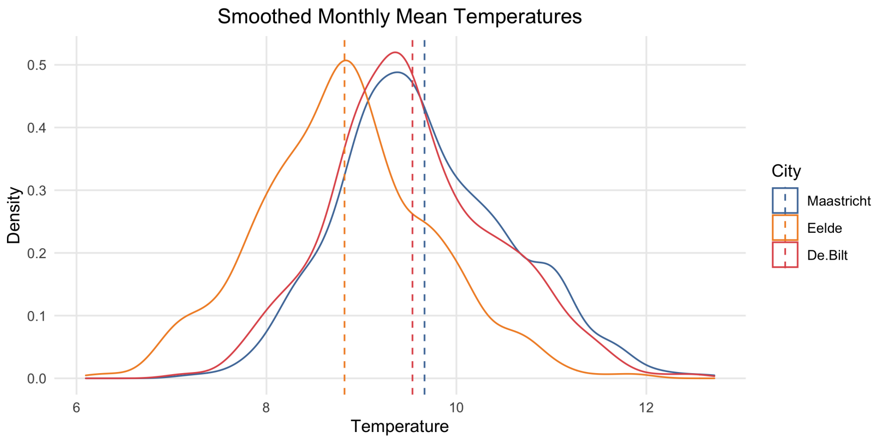


Figure 2.6: back to section 1.1.1

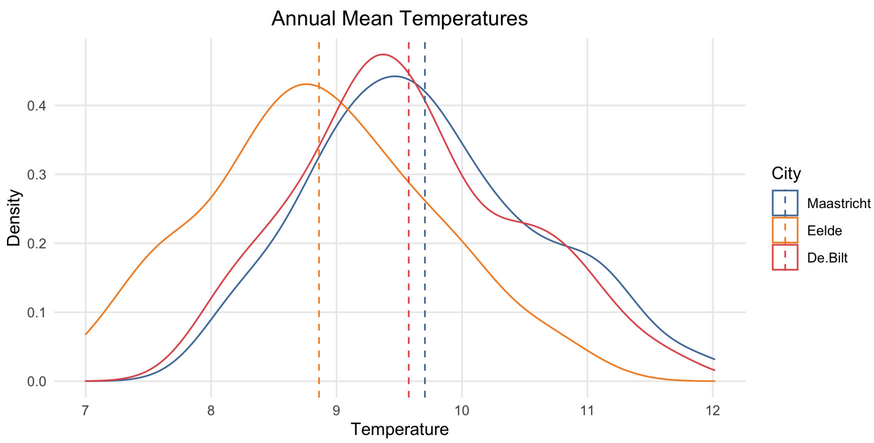


Figure 2.7: back to section 1.2.6

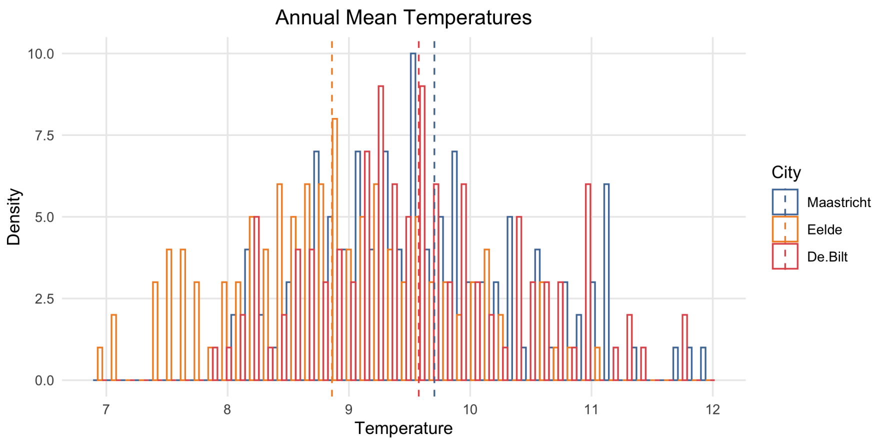


Figure 2.8: back to section 1.1.1

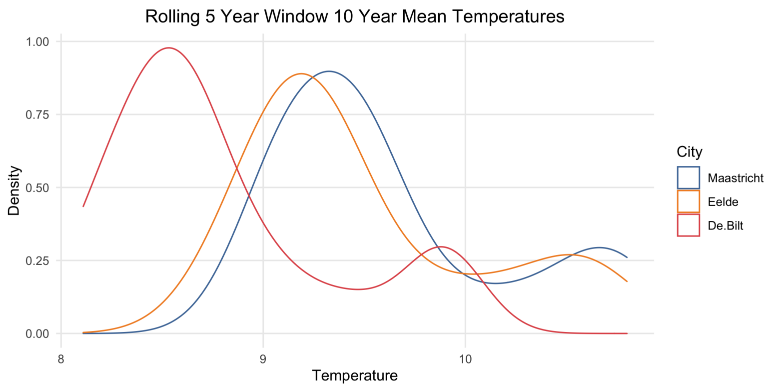


Figure 2.9: back to section 1.1.1

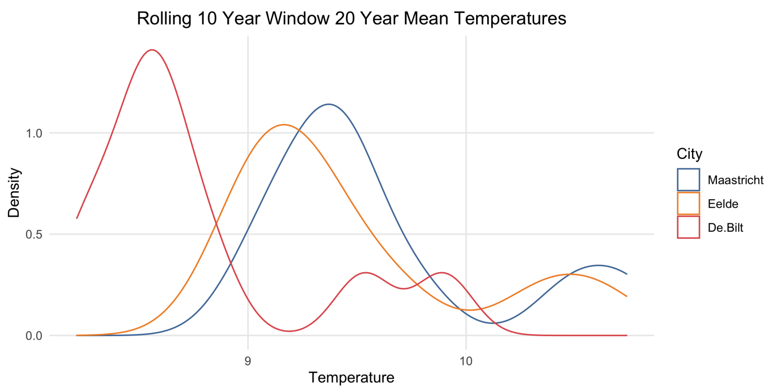


Figure 2.10: back to section 1.1.1

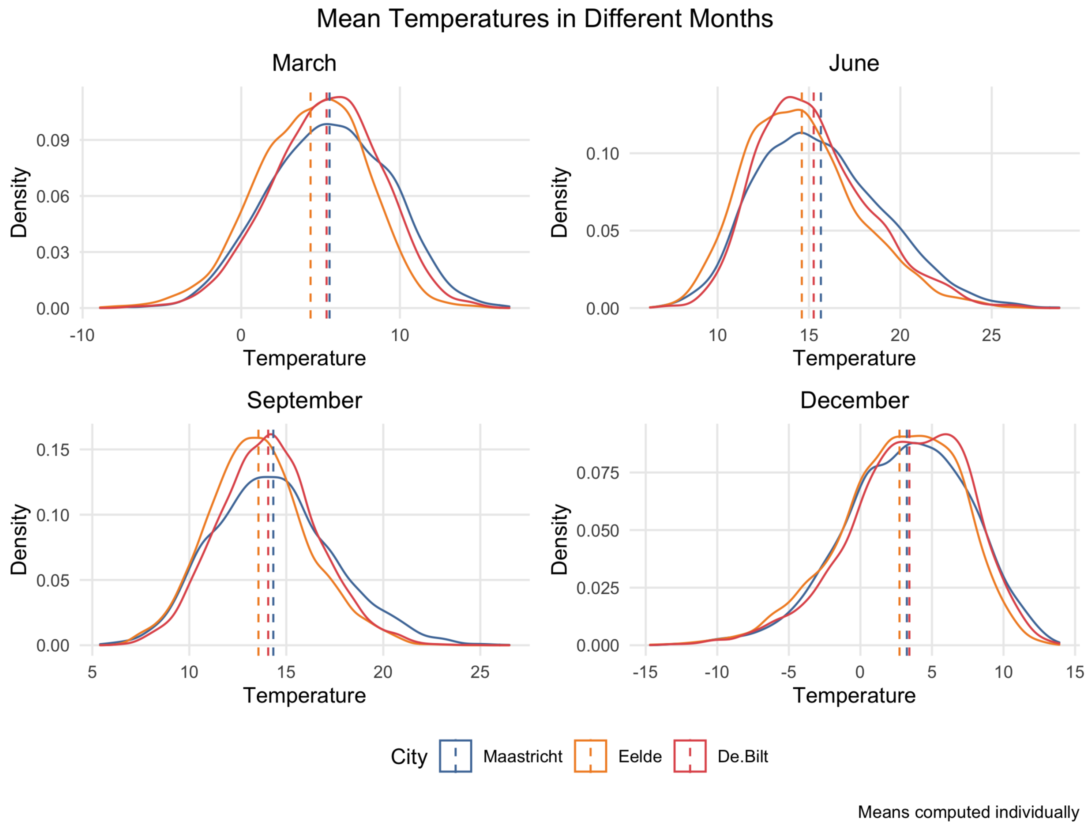


Figure 2.11: back to section 1.1.1

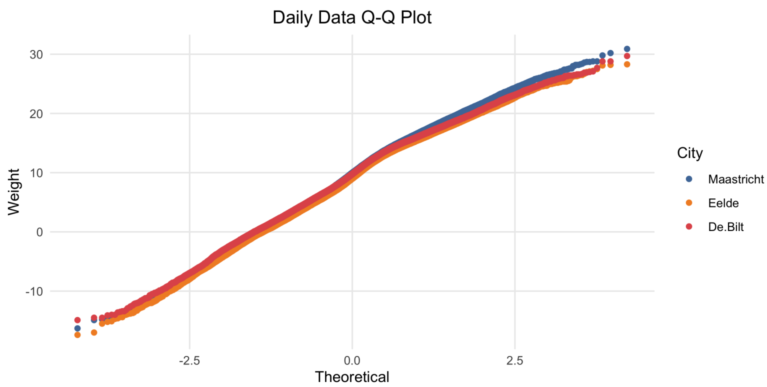


Figure 2.12: back to section 1.1.1

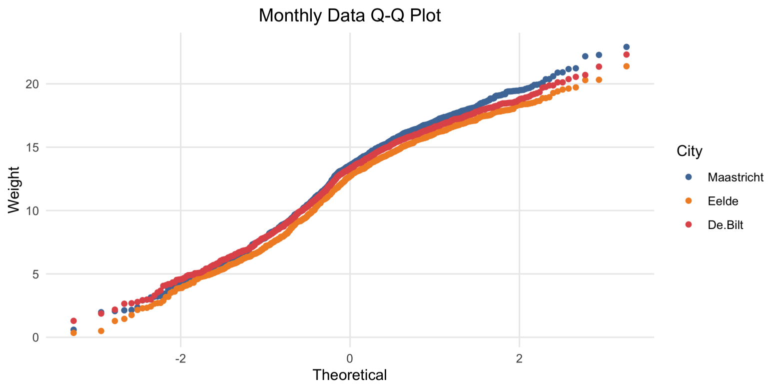


Figure 2.13: back to section 1.1.1

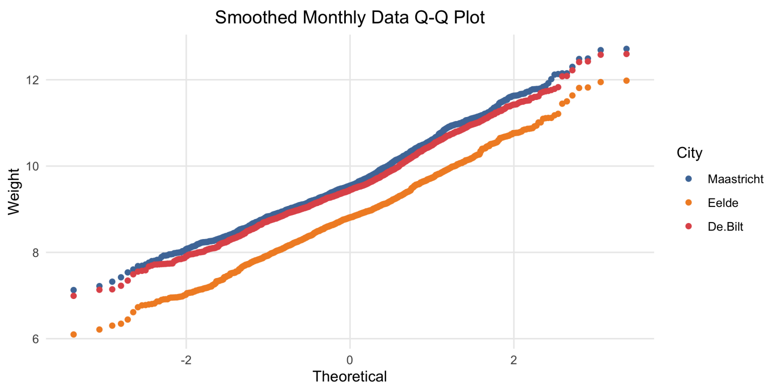


Figure 2.14: back to section 1.1.1

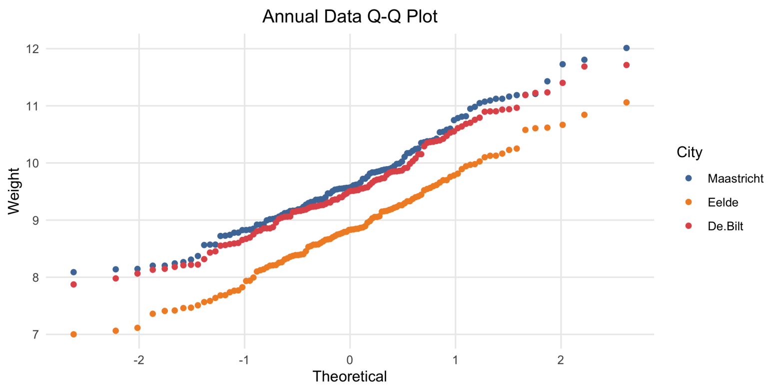


Figure 2.15: back to section 1.3.1

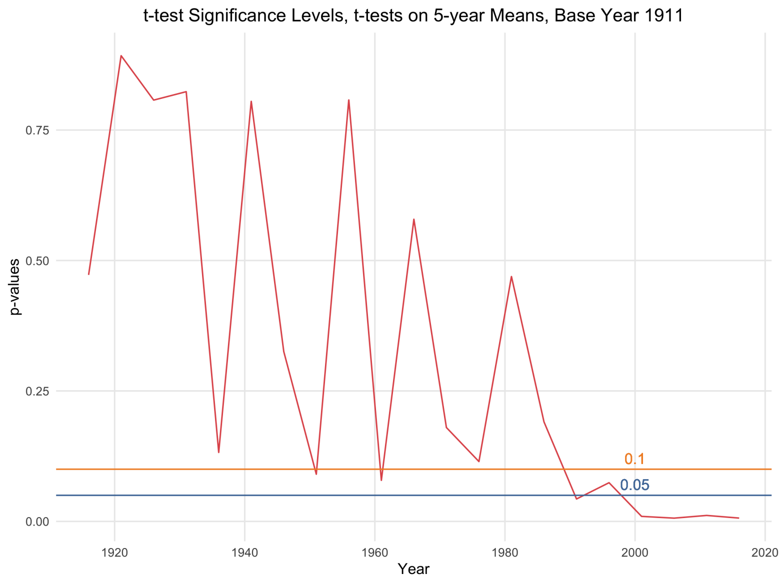


Figure 2.16: back to section 1.4.2

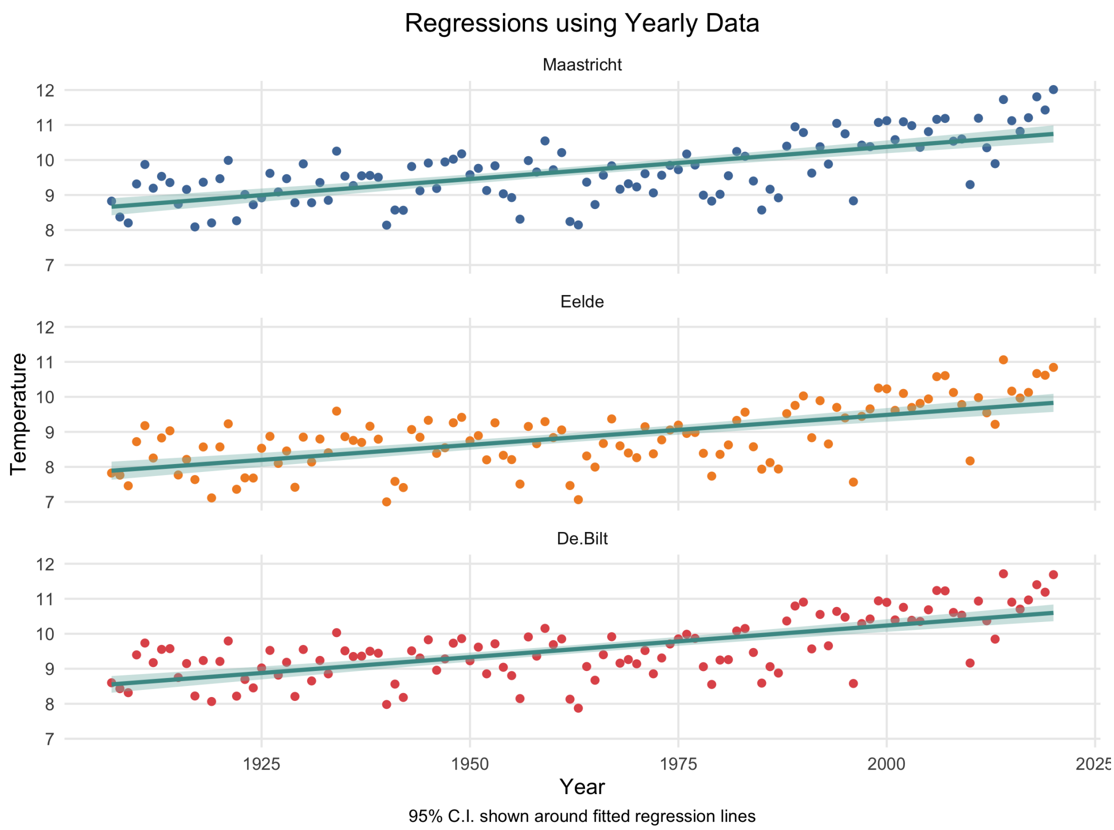


Figure 2.18: back to section 1.4.2

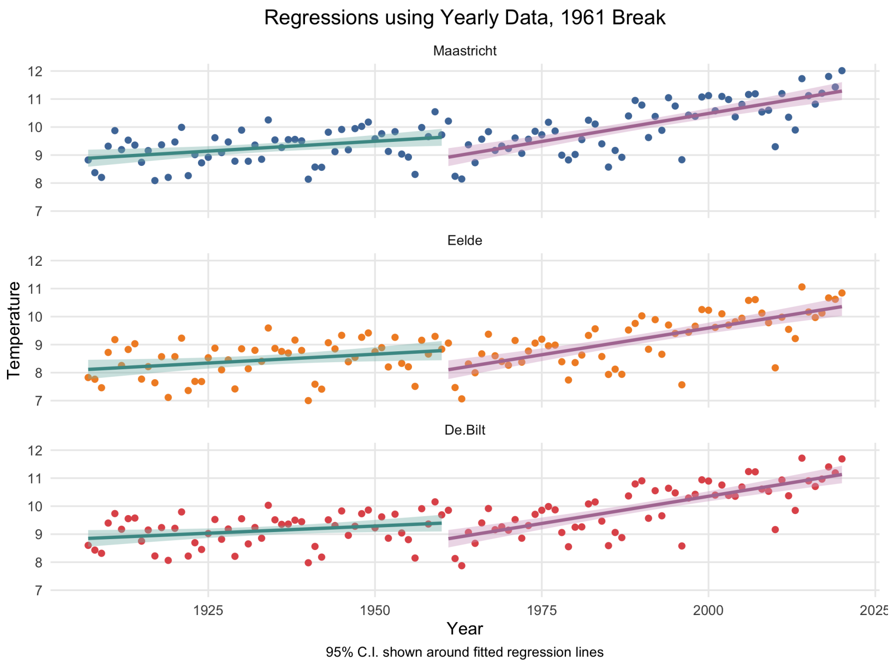


Figure 2.19: back to section 1.4.2

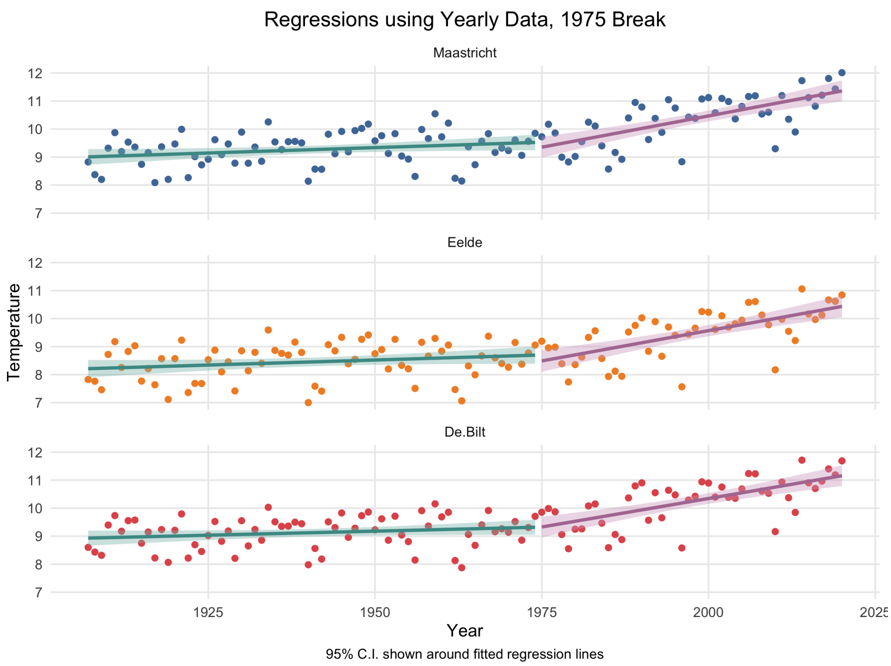


Figure 2.20: back to section 1.4.2

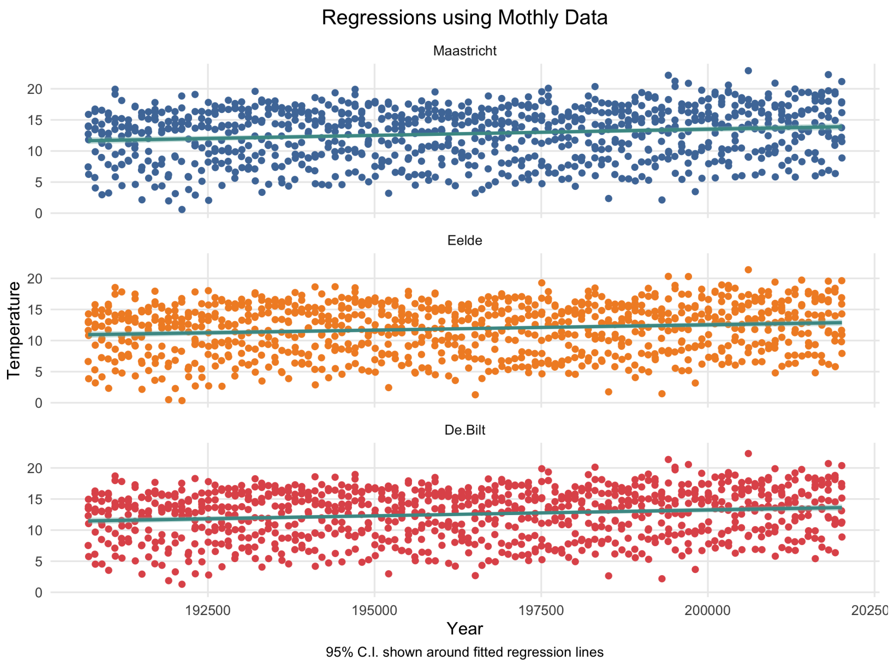


Figure 2.21: back to section 1.4.2

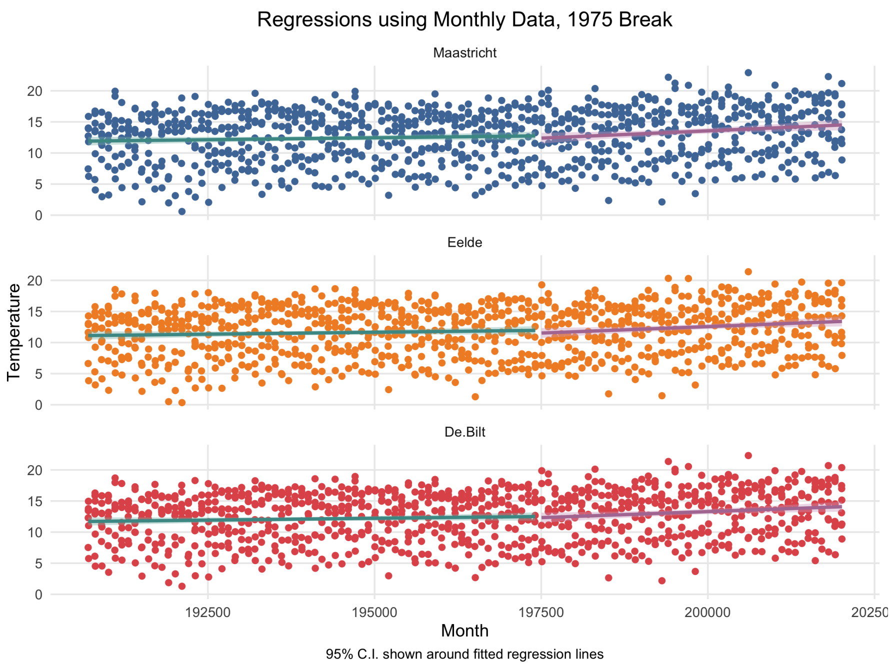


Figure 2.22: back to section 1.2.6

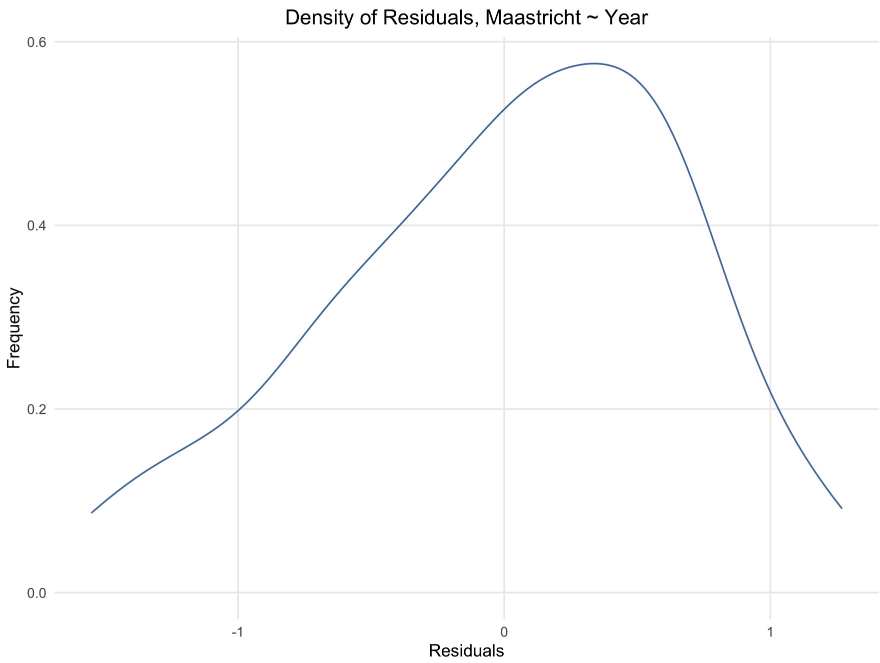
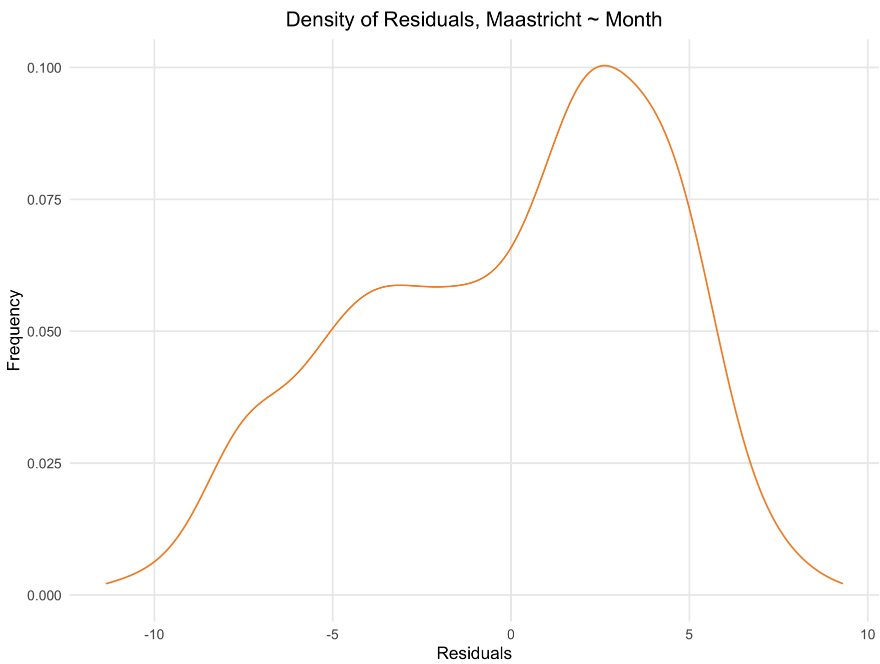


Figure 2.23: back to section 1.2.6



2.2 Appendix B: Tables

back to section 1.1.1

Table 2.1: Daily Data

Statistic	N	Mean	St. Dev.	Min	Max
De Bilt	41,639	9.576	6.299	−14.900	29.700
Eelde	41,639	8.860	6.411	−17.400	28.300
Maastricht	41,639	9.705	6.665	−16.300	30.900

Table 2.2: Monthly Data

Statistic	N	Mean	St. Dev.	Min	Max
De Bilt	912	12.553	4.004	1.293	22.310
Eelde	912	11.912	4.065	0.343	21.387
Maastricht	912	12.781	4.245	0.593	22.913

Table 2.3: Smoothed Monthly Data

Statistic	N	Mean	St. Dev.	Min	Max
De Bilt	1,356	9.537	0.885	6.992	12.598
Eelde	1,356	8.822	0.910	6.097	11.979
Maastricht	1,356	9.664	0.898	7.127	12.716

Table 2.4: Annual Data

Statistic	N	Mean	St. Dev.	Min	Max
De Bilt	114	9.576	0.878	7.874	11.714
Eelde	114	8.860	0.899	7.001	11.060
Maastricht	114	9.705	0.896	8.089	12.012

back to section 1.1.1 back to section 1.3.1

Table 2.5: t-tests, 1975 Break

	t-Statistic	p-value	C.I. Lower	C.I. Upper
De Bilt	7.5057	0.0000000	0.7923	Inf
Eelde	5.7760	0.0000004	0.6173	Inf
Maastricht	7.0249	0.0000000	0.7513	Inf

Table 2.6: t-tests, 10-Year Means

	Base Year	Comparison Year	t-Statistic	p-value	C.I. Lower	C.I. Upper
1	1916	1926	0.4528	0.67418	-0.8467	1.1765
2	1916	1936	-0.5522	0.61030	-1.1578	0.7741
3	1916	1946	-0.2448	0.81872	-1.0471	0.8777
4	1916	1956	-0.8550	0.44074	-1.2860	0.6805
5	1916	1966	-0.5584	0.60700	-1.3018	0.8690
6	1916	1976	-1.3613	0.24545	-1.4283	0.4899
7	1916	1986	-0.7651	0.48694	-1.2959	0.7363
8	1916	1996	-2.3449	0.08290	-2.1365	0.2055
9	1916	2006	-4.4863	0.01100	-2.6658	-0.6264
10	1916	2016	-4.3321	0.01235	-2.5738	-0.5630

Table 2.7: t-tests, 10-Year Medians

	Base Year	Comparison Year	t-Statistic	p-value	C.I. Lower	C.I. Upper
1	1916	1926	0.5142	0.63477	-0.9429	1.3672
2	1916	1936	-0.4863	0.65486	-1.2951	0.9233
3	1916	1946	-0.7701	0.49139	-1.3883	0.8190
4	1916	1956	-0.7738	0.48231	-1.5315	0.8643
5	1916	1966	-0.8580	0.43964	-1.5391	0.8143
6	1916	1976	-1.1322	0.32770	-1.5387	0.6770
7	1916	1986	-0.3649	0.73417	-1.2940	0.9964
8	1916	1996	-2.7803	0.04981	-2.4614	-0.0017
9	1916	2006	-3.6657	0.02154	-2.7861	-0.3837
10	1916	2016	-4.0551	0.01771	-2.7079	-0.4693

Table 2.8: t-tests, 5-Year Means

	Base Year	Comparison Year	t-Statistic	p-value	C.I. Lower	C.I. Upper
1	1911	1916	-0.7936	0.47237	-1.2840	0.7153
2	1911	1921	-0.1441	0.89243	-1.0082	0.9088
3	1911	1926	0.2608	0.80735	-0.9256	1.1158
4	1911	1931	-0.2388	0.82351	-1.1711	0.9887
5	1911	1936	-1.9212	0.13220	-1.4347	0.2810
6	1911	1941	-0.2640	0.80493	-1.0713	0.8856
7	1911	1946	-1.1218	0.32536	-1.2581	0.5362
8	1911	1951	-2.2312	0.09044	-1.8094	0.2020
9	1911	1956	-0.2602	0.80761	-1.0067	0.8343
10	1911	1961	-2.3975	0.07870	-2.0533	0.1765
11	1911	1966	0.6041	0.57903	-0.8046	1.2472
12	1911	1971	-1.6255	0.17986	-1.4336	0.3766
13	1911	1976	-2.0117	0.11469	-1.6529	0.2645
14	1911	1981	-0.7998	0.46918	-1.2921	0.7165
15	1911	1986	-1.5759	0.19066	-1.5387	0.4263
16	1911	1991	-2.9908	0.04295	-2.2389	-0.0605
17	1911	1996	-2.5124	0.07416	-2.3078	0.1764
18	1911	2001	-4.7152	0.00962	-2.7308	-0.6975
19	1911	2006	-5.2894	0.00624	-2.8424	-0.8825
20	1911	2011	-4.4625	0.01142	-2.5921	-0.5984
21	1911	2016	-5.2507	0.00635	-2.7925	-0.8593

back to section 1.3.1

back to section 1.1.2

Table 2.9: Structural Break in Yearly Data

	City	F-Statistic	p-value
1	De Bilt	109.18424	0.00000
2	Eelde	123.99164	0.00000
3	Maastricht	116.53588	0.00000

Table 2.10: Structural Break in Monthly Data

	City	F-Statistic	p-value
1	De Bilt	1341.99357	0.00000
2	Eelde	1350.22066	0.00000
3	Maastricht	1347.22753	0.00000

Table 2.11: Structural Break Breakpoints

	City	Yearly Data Breakpoint	Monthly Data Breakpoint
1	De Bilt	1961	1963.09
2	Eelde	1961	1963.09
3	Maastricht	1961	1964.09

back to section 1.1.2

back to section 1.3.1

Table 2.12: Paired t-tests, Yearly Data, 1961 Break

	t-Statistic	p-value	C.I. Lower	C.I. Upper
De Bilt	6.715	0.0000000	0.5298	0.9808
Eelde	5.741	0.0000004	0.4397	0.9115
Maastricht	6.453	0.0000000	0.4931	0.9376

Table 2.13: Paired t-tests, Monthly Data, 1961 Break

	t-Statistic	p-value	C.I. Lower	C.I. Upper
De Bilt	6.715	0.0000000	0.5298	0.9808
Eelde	5.741	0.0000004	0.4397	0.9115
Maastricht	6.453	0.0000000	0.4931	0.9376

Table 2.14: Paired t-tests, Yearly Data, 1975 Break

	t-Statistic	p-value	C.I. Lower	C.I. Upper
De Bilt	7.506	0.0000000	0.7467	1.2949
Eelde	5.776	0.0000007	0.5668	1.1742
Maastricht	7.025	0.0000000	0.7042	1.2707

Table 2.15: Paired t-tests, Monthly Data, 1975 Break

	t-Statistic	p-value	C.I. Lower	C.I. Upper
De Bilt	9.337	0.0000000	0.7043	1.0801
Eelde	7.168	0.0000000	0.5068	0.8900
Maastricht	8.325	0.0000000	0.6816	1.1031

back to section 1.3.1

back to section 1.3.2

Table 2.16: F-tests, Yearly Data, 1961 Break

	F-Statistic	p-value	C.I. Lower	C.I. Upper	Point Est.
De Bilt	0.43077	0.00239	0.2512572	0.7386	0.4308
Eelde	0.54814	0.02906	0.3197109	0.9398	0.5481
Maastricht	0.48549	0.00892	0.2831702	0.8324	0.4855

Table 2.17: F-tests, Monthly Data, 1961 Break

	F-Statistic	p-value	C.I. Lower	C.I. Upper	Point Est.
De Bilt	0.94190	0.52764	0.7821529	1.1343	0.9419
Eelde	1.00639	0.94642	0.8357017	1.2119	1.0064
Maastricht	0.95543	0.63038	0.7933830	1.1506	0.9554

Table 2.18: F-tests, Yearly Data, 1961 Break

	F-Statistic	p-value	C.I. Lower	C.I. Upper	Point Est.
De Bilt	0.431	0.0023937	0.2513	0.7386	0.4308
Eelde	0.548	0.0290574	0.3197	0.9398	0.5481
Maastricht	0.485	0.0089244	0.2832	0.8324	0.4855

back to section 1.3.2

Table 2.19: F-tests, Monthly Data, 1961 Break

	F-Statistic	p-value	C.I. Lower	C.I. Upper	Point Est.
De Bilt	0.942	0.5276390	0.7822	1.1343	0.9419
Eelde	1.006	0.9464201	0.8357	1.2119	1.0064
Maastricht	0.955	0.6303802	0.7934	1.1506	0.9554

Table 2.20: F-tests, Yearly Data, 1975 Break

	F-Statistic	p-value	C.I. Lower	C.I. Upper	Point Est.
De Bilt	0.462	0.0118367	0.2538	0.8405	0.4619
Eelde	0.509	0.0272475	0.2796	0.9258	0.5088
Maastricht	0.440	0.0075549	0.2418	0.8006	0.4400

Table 2.21: F-tests, Monthly Data, 1975 Break

	F-Statistic	p-value	C.I. Lower	C.I. Upper	Point Est.
De Bilt	0.934	0.5153017	0.7612	1.1468	0.9343
Eelde	0.972	0.7837219	0.7917	1.1928	0.9717
Maastricht	0.914	0.3900880	0.7447	1.1220	0.9141

back to section 1.3.2 back to section 1.4.1

Table 2.22: Manually Computed Regression Coefficients, Maastricht, Yearly Data

	estimate	se	p-value
alpha	-26.427605	3.693529	0.00000000
beta	0.018402	0.001881	0.00000000

Table 2.23: Manually Computed Regression Coefficients, De Bilt, Yearly Data

	estimate	se	p-value
alpha	-25.916052	3.608964	0.00000000
beta	0.018076	0.001838	0.00000000

Table 2.24: Manually Computed Regression Coefficients, Eelde, Yearly Data

	estimate	se	p-value
alpha	-24.809977	3.915264	0.00000001
beta	0.017148	0.001994	0.00000000

back to section 1.4.1

Table 2.25: Manually Computed Regression Coefficients, Maastricht, Monthly Data

	estimate	se	p-value
alpha	-26.252276	8.292907	0.00159894
beta	0.000199	0.000042	0.00000290

Table 2.26: Manually Computed Regression Coefficients, De Bilt, Monthly Data

	estimate	se	p-value
alpha	-24.628598	7.820923	0.00169136
beta	0.000189	0.000040	0.00000231

Table 2.27: Manually Computed Regression Coefficients, Eelde, Monthly Data

	estimate	se	p-value
alpha	-21.659638	7.959094	0.00662534
beta	0.000171	0.000041	0.00002704

back to section 1.4.1 back to section 1.4.1

Table 2.28: Regressions, Yearly Data

	Dependent variable:		
	de_bilt	eelde	maastricht
	(1)	(2)	(3)
year	0.018***	0.017***	0.018***
	(0.002)	(0.002)	(0.002)
Constant	−25.916***	−24.810***	−26.428***
	(3.609)	(3.915)	(3.694)
Observations	114	114	114
R^2	0.463	0.398	0.461
Adjusted R^2	0.459	0.392	0.456
Residual Std. Error (df = 112)	0.646	0.701	0.661
F Statistic (df = 1; 112)	96.741***	73.977***	95.726***
Note:	*p<0.1; **p<0.05; ***p<0.01		

Table 2.29: Regressions, Monthly Data

	Dependent variable:		
	de_bilt	eelde	maastricht
	(1)	(2)	(3)
month	0.0002***	0.0002***	0.0002***
	(0.00004)	(0.00004)	(0.00004)
Constant	−24.629***	−21.660***	−26.252***
	(7.821)	(7.959)	(8.293)
Observations	912	912	912
R^2	0.024	0.019	0.024
Adjusted R^2	0.023	0.018	0.023
Residual Std. Error (df = 910)	3.958	4.028	4.197
F Statistic (df = 1; 910)	22.608***	17.797***	22.160***
Note:	*p<0.1; **p<0.05; ***p<0.01		

back to section 1.4.1
back to section 1.4.2

Table 2.30: Regressions, Yearly Data, Before 1961 Break

	Dependent variable:		
	de_bilt	eelde	maastricht
	(1)	(2)	(3)
year	0.011**	0.013**	0.015***
	(0.005)	(0.005)	(0.005)
Constant	−12.408	−16.970	−20.068**
	(9.003)	(10.369)	(9.247)
Observations	55	55	55
R^2	0.098	0.102	0.160
Adjusted R^2	0.080	0.085	0.144
Residual Std. Error (df = 53)	0.548	0.631	0.563
F Statistic (df = 1; 53)	5.726**	6.016**	10.072***
Note:			*p<0.1; **p<0.05; ***p<0.01

back to section 1.4.2

Table 2.31: Regressions, Yearly Data, After 1961 Break

	Dependent variable:		
	de_bilt	eelde	maastricht
	(1)	(2)	(3)
year	0.039***	0.038***	0.039***
	(0.005)	(0.006)	(0.005)
Constant	−67.334***	−65.982***	−67.962***
	(10.392)	(11.159)	(10.344)
Observations	55	55	55
R^2	0.510	0.461	0.517
Adjusted R^2	0.501	0.451	0.508
Residual Std. Error (df = 53)	0.615	0.660	0.612
F Statistic (df = 1; 53)	55.230***	45.317***	56.794***

Note: *p<0.1; **p<0.05; ***p<0.01

back to section 1.4.2

Table 2.32: Regressions, Yearly Data, Before 1975 Break

	Dependent variable:		
	de_bilt	eelde	maastricht
	(1)	(2)	(3)
year	0.003	0.001	0.003
	(0.007)	(0.007)	(0.007)
Constant	2.459	7.338	3.046
	(12.849)	(14.366)	(13.182)
Observations	45	45	45
R^2	0.006	0.0002	0.005
Adjusted R^2	−0.017	−0.023	−0.018
Residual Std. Error (df = 43)	0.573	0.641	0.588
F Statistic (df = 1; 43)	0.278	0.008	0.231
Note:	*p<0.1; **p<0.05; ***p<0.01		

back to section 1.4.2

Table 2.33: Regressions, Yearly Data, After 1975 Break

	Dependent variable:		
	de_bilt	eelde	maastricht
	(1)	(2)	(3)
year	0.042***	0.045***	0.046***
	(0.007)	(0.008)	(0.007)
Constant	−74.214***	−81.196***	−81.052***
	(14.511)	(15.279)	(14.872)
Observations	45	45	45
R^2	0.441	0.450	0.468
Adjusted R^2	0.428	0.437	0.455
Residual Std. Error (df = 43)	0.633	0.666	0.648
F Statistic (df = 1; 43)	33.882***	35.212***	37.789***

Note: *p<0.1; **p<0.05; ***p<0.01

back to section 1.4.2
back to section 1.4.2

Table 2.34: Regressions, Monthly Data, Before 1961 Break

	Dependent variable:		
	de_bilt	eelde	maastricht
	(1)	(2)	(3)
month	0.0002*	0.0002*	0.0002*
	(0.0001)	(0.0001)	(0.0001)
Constant	−24.529	−27.514	−28.722
	(22.191)	(22.942)	(23.610)
Observations	447	447	447
R^2	0.006	0.006	0.007
Adjusted R^2	0.004	0.004	0.005
Residual Std. Error (df = 445)	3.911	4.044	4.162
F Statistic (df = 1; 445)	2.724*	2.896*	3.022*

Note: *p<0.1; **p<0.05; ***p<0.01

back to section 1.4.2

Table 2.35: Regressions, Monthly Data, After 1961 Break

	Dependent variable:		
	de_bilt	eelde	maastricht
	(1)	(2)	(3)
month	0.0004***	0.0004***	0.0004***
	(0.0001)	(0.0001)	(0.0001)
Constant	−64.815***	−62.279***	−72.960***
	(23.303)	(23.336)	(24.595)
Observations	447	447	447
R^2	0.024	0.022	0.027
Adjusted R^2	0.022	0.020	0.025
Residual Std. Error (df = 445)	3.993	3.999	4.214
F Statistic (df = 1; 445)	11.143***	10.204***	12.270***

Note: *p<0.1; **p<0.05; ***p<0.01

back to section 1.4.2

Table 2.36: Regressions, Monthly Data, Before 1975 Break

	Dependent variable:		
	de_bilt	eelde	maastricht
	(1)	(2)	(3)
month	−0.00001	−0.00004	−0.00004
	(0.0002)	(0.0002)	(0.0002)
Constant	13.640	19.594	20.241
	(29.861)	(30.492)	(31.409)
Observations	368	368	368
R^2	0.00001	0.0002	0.0002
Adjusted R^2	−0.003	−0.003	−0.003
Residual Std. Error (df = 366)	3.897	3.979	4.099
F Statistic (df = 1; 366)	0.002	0.066	0.060
Note:			*p<0.1; **p<0.05; ***p<0.01

back to section 1.4.2

Table 2.37: Regressions, Monthly Data, After 1975 Break

	Dependent variable:		
	de_bilt	eelde	maastricht
	(1)	(2)	(3)
month	0.0004**	0.0004***	0.0005***
	(0.0002)	(0.0002)	(0.0002)
Constant	−64.522**	−70.539**	−80.383**
	(31.360)	(31.366)	(33.269)
Observations	368	368	368
R^2	0.017	0.019	0.021
Adjusted R^2	0.014	0.016	0.019
Residual Std. Error (df = 366)	3.998	3.999	4.241
F Statistic (df = 1; 366)	6.144**	7.003***	7.955***
Note:			*p<0.1; **p<0.05; ***p<0.01

back to section 1.4.2

Table 2.38: Regressions, Yearly Data, Before and After 1975 Break

	Dependent variable:	
	maastricht Year < 1975	maastricht Year > 1975
	(1)	(2)
year	0.003	
	(0.007)	
year		0.046^{***}
		(0.007)
Constant	3.046	-81.052^{***}
	(13.182)	(14.872)
Observations	45	45
R^2	0.005	0.468
Adjusted R^2	−0.018	0.455
Residual Std. Error (df = 43)	0.588	0.648
F Statistic (df = 1; 43)	0.231	37.789^{***}
Note:		*p<0.1; **p<0.05; ***p<0.01

back to section 1.4.2
back to section 1.2.4

Table 2.39: White Tests for Heteroskedasticity

	Test Statistic	p-value
De Bilt, Yearly Data	1.38175	0.50114
Eelde, Yearly Data	0.69513	0.70641
Maastricht, Yearly Data	2.67000	0.26316
De Bilt, Monthly Data	0.03966	0.98037
Eelde, Monthly Data	0.62253	0.73252
Maastricht, Monthly Data	0.29890	0.86118

Table 2.40: White Tests for Heteroskedasticity

	Test Statistic	p-value
De Bilt, before 1975	1.56871	0.45641
Eelde, before 1975	1.15832	0.56037
Maastricht, before 1975	3.00716	0.22233
De Bilt, after 1975	0.37004	0.83109
Eelde, after 1975	0.04104	0.97969
Maastricht, after 1975	0.04971	0.97545

back to section 1.2.4

back to section 1.5

Table 2.41: Bootstrap: t-test for Regression Coefficients

	Q^*	CI lower	CI upper
De Bilt, Yearly Data, Pairs	0.1369	-0.356699	0.397091
De Bilt, Yearly Data, Residuals		-0.712420	0.736694
Eelde, Yearly Data, Pairs	0.1405	-0.295937	0.339347
Eelde, Yearly Data, Residuals		-0.540706	0.731274
Maastricht, Yearly Data, Pairs	0.1357	-0.356233	0.401201
Maastricht, Yearly Data, Residuals		-0.704909	0.731274
De Bilt, Monthly Data, Pairs	0.1369	-0.107326	0.202439
De Bilt, Monthly Data, Residuals		-0.057493	0.058101
Eelde, Monthly Data, Pairs	0.1405	-0.139981	0.213268
Eelde, Monthly Data, Residuals		-0.066346	0.055207
Maastricht, Monthly Data, Pairs	0.1357	-0.110383	0.197385
Maastricht, Monthly Data, Residuals		-0.057359	0.055207

Table 2.42: Bootstrap: t-test for Regression Coefficients

	Q^*	CI lower	CI upper
De Bilt, Yearly Data, Pairs	0.1369	-0.356699	0.397091
De Bilt, Yearly Data, Residuals		-0.712420	0.736694
Eelde, Yearly Data, Pairs	0.1405	-0.295937	0.339347
Eelde, Yearly Data, Residuals		-0.540706	0.731274
Maastricht, Yearly Data, Pairs	0.1357	-0.356233	0.401201
Maastricht, Yearly Data, Residuals		-0.704909	0.731274
De Bilt, Monthly Data, Pairs	0.1369	-0.107326	0.202439
De Bilt, Monthly Data, Residuals		-0.057493	0.058101
Eelde, Monthly Data, Pairs	0.1405	-0.139981	0.213268
Eelde, Monthly Data, Residuals		-0.066346	0.055207
Maastricht, Monthly Data, Pairs	0.1357	-0.110383	0.197385
Maastricht, Monthly Data, Residuals		-0.057359	0.055207

back to section 1.5

2.3 Appendix C: Source Code (Main Script)

```r
#########################HOUSEKEEPING########################

rm(list = ls(all = TRUE)) ###CLEAR ALL
# Package names
packages <- c("data.table", "dplyr", "zoo", "tidyr", "ggplot2", "ggthemes", "scales", "strucchange", "xts",
              "skedastic", "tidyverse", "xtable", "knitr", "stargazer", "patchwork", "remotes", "broom", "purrr")
# package grateful must be installed by hand# install.packages("remotes")
remotes::install_github("Pakillo/grateful")
# Install packages not yet installed
installed_packages <- packages %in% rownames(installed.packages())
if (any(installed_packages == FALSE)) {
  install.packages(packages[!installed_packages])
}

#load packages
invisible(lapply(packages, library, character.only = TRUE))


Paths = c("/Users/ts/Git/Mathematical-Stats", "Users/chumasharajapakshe/Documents/GitHub/Mathematical-Stats")
names(Paths) = c("ts", "chumasharajapakshe")
setwd(Paths[Sys.info()[7]])

#########################DATA IMPORT########################

#annual
da <- fread('./Data/AnnualTemp.csv', dec=",")
colnames(da) <- c('year', 'de_bilt', 'eelde', 'maastricht')
#monthly
dm <- fread('./Data/MonthlyTemp.csv', dec=",")
colnames(dm) <- c('month', 'de_bilt', 'eelde', 'maastricht')

#smoothed
dms <- fread('./Data/SMTemp.csv', dec=",")
colnames(dms) <- c('month', 'de_bilt', 'eelde', 'maastricht')

#daily
dd <- fread('./Data/DailyTemp.csv', dec=",")
colnames(dd) <- c('date', 'de_bilt', 'eelde', 'maastricht')


#########################DATA ANALYSIS########################
# the following functions all work the same way:
# 1. They first subset to a new data.table the annual/monthly/monthly (smoothed)/daily
# data by applying a user-specified function (mean, median, sum, etc.) in a
```

```
# rolling window specified by the user
# 2. They then subset the original data to a second new table and generate row numbers
# and keep Years/months etc. and bind these 2 "new" tables together
# 3. They then drop all rows outside the specified time window so that we are left with
# a subset of data with e.g. 10 year means or 6 day medians
# We use these to experiment with different subsets
#functions
{
  xYearStat <- function(x, STATNAME){
    #generate data.table with rolling means, window=x
    xYrS <- as.data.table(frollapply(da[, .(de_bilt, eelde, maastricht)], x, FUN = STATNAME))[, ID := .I ]
    dateSet <- da[, ID := .I][, .(year, ID)]
    xYrStatInclYr <- cbind(xYrS, dateSet)
    xYST <- xYrStatInclYr[ID %% x == 0 , .(Year = year, Eelde = V1, 'De Bilt' = V2, Maastricht = V3)]
    return(xYST)
  }

  xYearYoverlapStat <- function(x, y, STATNAME){
    #generate data.table with rolling means, window=x
    xYrS <- as.data.table(frollapply(da[, .(de_bilt, eelde, maastricht)], x, FUN = STATNAME))[, ID := .I ]
    dateSet <- da[, ID := .I][, .(year, ID)]
    xYrStatInclYr <- cbind(xYrS, dateSet)
    z = x - y
    xYST <- xYrStatInclYr[ID %% z == 0 , .(Year = year, Eelde = V1, 'De.Bilt' = V2, Maastricht = V3)]
    return(xYST)
  }

  xMonthStat <- function(x, STATNAME){
    xMosRS <- as.data.table(frollapply(dm[, .(de_bilt, eelde, maastricht)], x, FUN = STATNAME))[, ID := .I ]
    dateSet <- dm[, ID := .I][, .(month, ID)]
    xMosRSInclMonth <- cbind(xMosRS, dateSet)
    xMosStat <- xMosRSInclMonth[(ID) %% x == 0 , .(Month = month , Eelde = V1, 'De Bilt' = V2, Maastricht = V3)]
    return(xMosStat)
  }

  xMonthSmoothedStat <- function(x, STATNAME){
    xMosSRS <- as.data.table(frollapply(dms[, .(de_bilt, eelde, maastricht)], x, FUN = STATNAME))[, ID := .I ]
    dateSet <- dms[, ID := .I][, .(month, ID)]
    xMosSRSInclMonth <- cbind(xMosSRS, dateSet)
    xMosSStat <- xMosSRSInclMonth[(ID) %% x == 0 , .(Month = month , Eelde = V1, 'De Bilt' = V2, Maastricht = V3)]
    return(xMosSStat)
```

```
}

xDayStat <- function(x, STATNAME){
  xDayRS <- as.data.table(frollapply(dd[, .(de_bilt, eelde, maastricht)], x, FUN = STATNAME))[, ID := .I ]
  dateSet <- dd[, ID := .I][, .(date, ID)]
  xDayRSInclDate <- cbind(xDayRS, dateSet)
  xMosSStat <- xDayRSInclDate[(ID) %% x == 0 , .(Date = date, Eelde = V1, 'De Bilt' = V2, Maastricht = V3)]
  return(xMosSStat)
}

subsetMonth <- function(mm){
  mm <- str_pad(as.character(mm), 2, side = "left", pad = '0')

  subset <- dd[, DateCol := as.Date(as.character(date), format = "%Y%m%d")
  ][, month := format(as.Date(DateCol), "%m")]
  compSet <- subset[month == mm, .(DateCol, eelde, de_bilt, maastricht)]
  return(compSet)
}

subsetMonths <- function(mmstart, mmend){
  mmstart <- str_pad(as.character(mmstart), 2, side = "left", pad = '0')
  mmend <- str_pad(as.character(mmstart), 2, side = "left", pad = '0')

  subset <- dd[, DateCol := as.Date(as.character(date), format = "%Y%m%d")
  ][, month := format(as.Date(DateCol), "%m")]
  compSet <- subset[month >= mmstart & month <=mmend, .(DateCol, eelde, de_bilt, maastricht)]
  return(compSet)
}

subsetMonthLong <- function(mm){
  mm <- str_pad(as.character(mm), 2, side = "left", pad = '0')

  subset <- dd[, DateCol := as.Date(as.character(date), format = "%Y%m%d")
  ][, month := format(as.Date(DateCol), "%m")]
  compSet <- subset[month == mm, .(Month = month, Eelde = eelde, De.Bilt = de_bilt, Maastricht = maastricht)]
  ret <- melt(compSet, id.vars = "Month", measure.vars = c("Maastricht", "Eelde", "De.Bilt"),
              variable.factor = T, variable.name = "City", value.name = "Temperature")[, Citymean := mean(Temperature), by = City]
  return(ret)
}

subsetMonthsLong <- function(mmstart,mmend){
  mmstart <- str_pad(as.character(mmstart), 2, side = "left", pad = '0')
  mmend <- str_pad(as.character(mmstart), 2, side = "left", pad = '0')

  subset <- dd[, DateCol := as.Date(as.character(date), format = "%Y%m%d")
```

```
    ][, month := format(as.Date(DateCol), "%m")]
    compSet <- subset[month >= mmstart & month <=mmend, .(Month = month, Eelde = eelde, De.Bilt = de_bilt, Maastricht = maastricht)]
    ret <- melt(compSet, id.vars = "Month", measure.vars = c("Maastricht", "Eelde", "De.Bilt"),
                variable.factor = T, variable.name = "City", value.name = "Temperature")[, Citymean := mean(Temperature), by = City]
    return(ret)
  }

  subsetDate <- function(mmdd){
    mmdd <- str_pad(as.character(mmdd), 4, side = "left", pad = '0')

    subset <- dd[, DateCol := as.Date(as.character(date), format = "%Y%m%d")
    ][, md := format(as.Date(DateCol), "%m%d")][, yd := format(as.Date(DateCol), "%Y%m%d")]

    compSet <- subset[mmdd == md, .(Date = md, eelde, de_bilt, maastricht, yd)]
    return(compSet)
  }

  subsetDateLong <- function(mmdd){
    mmdd <- str_pad(as.character(mmdd), 4, side = "left", pad = '0')

    subset <- dd[, DateCol := as.Date(as.character(date), format = "%Y%m%d")
    ][, md := format(as.Date(DateCol), "%m%d")]

    compSet <- subset[mmdd == md, .(Date = DateCol, Eelde = eelde, De.Bilt = de_bilt, Maastricht = maastricht)]
    ret <- melt(compSet, id.vars = "Date", measure.vars = c("Maastricht", "Eelde", "De.Bilt"),
                variable.factor = T, variable.name = "City", value.name = "Temperature")[, Citymean := mean(Temperature)]

    return(ret)
  }
}

#############PROBLEM##########
##############################
#last-minute (April 7, 22:02) realization
#that extension to monthly data explicitly
#requires deseasonalization
#leads to: plot(density(dm$maastricht)), which leads to
#the utterance of several expletives
test1 <- subsetMonths(3,10)
plot(density(test1$maastricht))
plot(density(test1$de_bilt))
test2 <- subsetMonths(4,11)
plot(density(test2$maastricht))
plot(density(test2$eelde))
test3 <- subsetMonths(4,11)
```

```
176 plot(density(test3$maastricht))
177 plot(density(test3$de_bilt))
178 plot(density(test3$eelde))
179
180 #test3 looks close enough to a normal distribution
181 #hail-mary:
182 #dm <- test3
183
184 ##SOLUTION
185 #dm <- test3
186 deseasonalize <- function(mmstart, mmend){
187
188   tsdat <- xts(dd, as.Date(as.character(dd$date), format = "%Y%m%d"))
189   ts_m = apply.monthly(tsdat, mean)
190   dmts <- as.data.table(ts_m)[,date := as.Date(index, format = "%m")][, month := format(date,
      ↪  "%m")][,mon:=as.numeric(month)]
191   dm2 <- cbind(dm$month, dmts$mon, dmts$de_bilt, dmts$eelde, dmts$maastricht)
192   colnames(dm2) <- c('month', 'mon', 'de_bilt', 'eelde', 'maastricht')
193   dm3 <- as.data.table(dm2)[mon >= 4 & mon <=11][, .(month, de_bilt, eelde, maastricht)]
194   colnames(dm3) <- c('month', 'de_bilt', 'eelde', 'maastricht')
195
196   return(dm3)
197 }
198
199 ############TO BE CLEAR, THE ADDITION OF
200 ##THIS FUNCTION AND THE CHANGE in 'dm' came at the
201 #VERY LAST MINUTE (APRIL 7 22:02), earlier results can very easily be re-obtained
202 ##BY COMMENTING OUT LINE 204
203 dmbackup <- dm
204 dm <- deseasonalize(4,11)
205
206 #gen datasets needed
207 {
208   #rolling window plots
209   rollingMean10_5 <- xYearYoverlapStat(10, 5, mean)
210   rollingMean20_10 <- xYearYoverlapStat(20, 10, mean)
211
212   # february <- subsetMonth(2)
213
214   meanTable10y <- xYearStat(10, mean)
215   meanTable5y <- xYearStat(5, mean)
216   meanTable50y <- xYearStat(50, mean)
217   medianTable10y <- xYearStat(10, median)
218   meanTable10mo <- xMonthStat(10, mean)
219   medianTable5mo <- xMonthStat(5, median)
220   meanTable20d <- xDayStat(20, mean)
221   varTable5y <- xYearStat(5, var)
222 }
223
```

```
#test for structural change
{
  structmat1 <- matrix(nrow = 3, ncol=2)
  rownames(structmat1) <- c('De Bilt', 'Eelde', 'Maastricht')
  colnames(structmat1) <- c('F-Statistic', 'p-value')

  scyM <- sctest(da$year ~ da$maastricht, type = "Chow")
  scyE <- sctest(da$year ~ da$eelde, type = "Chow")
  scyD <- sctest(da$year ~ da$de_bilt, type = "Chow")

  structmat1[3,1] <- scyM$statistic
  structmat1[3,2] <- scyM$p.value
  structmat1[2,1] <- scyE$statistic
  structmat1[2,2] <- scyE$p.value
  structmat1[1,1] <- scyD$statistic
  structmat1[1,2] <- scyD$p.value

  structtabY <- as.data.table(structmat1, keep.rownames = T)
  setnames(structtabY, "rn", "City")
  rm('scyM','scyE', 'scyD', 'structmat1')

  structmat2 <- matrix(nrow = 3, ncol=2)
  rownames(structmat2) <- c('De Bilt', 'Eelde', 'Maastricht')
  colnames(structmat2) <- c('F-Statistic', 'p-value')

  scmM <- sctest(dm$month ~ dm$maastricht, type = "Chow")
  scmE <- sctest(dm$month ~ dm$eelde, type = "Chow")
  scmD <- sctest(dm$month ~ dm$de_bilt, type = "Chow")

  structmat2[3,1] <- scmM$statistic
  structmat2[3,2] <- scmM$p.value
  structmat2[2,1] <- scmE$statistic
  structmat2[2,2] <- scmE$p.value
  structmat2[1,1] <- scmD$statistic
  structmat2[1,2] <- scmD$p.value

  structtabM <- as.data.table(structmat2, keep.rownames = T)
  setnames(structtabM, "rn", "City")

  rm('scmM','scmE', 'scmD', 'structmat2')

  #find breakpoints
  structmat3 <- matrix(nrow = 3, ncol=2)
  rownames(structmat3) <- c('De Bilt', 'Eelde', 'Maastricht')
  colnames(structmat3) <- c('Yearly Data Breakpoint', 'Monthly Data Breakpoint')

  ybpm <-  breakpoints(da$year ~ da$maastricht, h = 0.35, breaks = 1)
  structmat3[3,1] <- da[ID == ybpm$breakpoints, year]
  ybpe <- breakpoints(da$year ~ da$eelde, h = 0.35, breaks = 1)
```

```
  structmat3[2,1] <- da[ID == ybpe$breakpoints, year]
  ybpd <- breakpoints(da$year ~ da$de_bilt, h = 0.35, breaks = 1)
  structmat3[1,1] <- da[ID == ybpd$breakpoints, year]

  mbpm <-  breakpoints(dm$month ~ dm$maastricht, h = 0.35, breaks = 1)
  structmat3[3,2] <- dm[ID == mbpm$breakpoints, month]/100
  mbpe <- breakpoints(dm$month ~ dm$eelde, h = 0.35, breaks = 1)
  structmat3[2,2] <- dm[ID == mbpe$breakpoints, month]/100
  mbpd <- breakpoints(dm$month ~ dm$de_bilt, h = 0.35, breaks = 1)
  structmat3[1,2] <- dm[ID == mbpd$breakpoints, month]/100

  structtabBP <- as.data.table(structmat3, keep.rownames = T)
  setnames(structtabBP, "rn", "City")
  rm('ybpm','ybpe', 'ybpd', 'mbpd', 'mbpe', 'structmat3')
}

#subset according to breakpoint results
prebreakY <- da[year <= 1961]
postbreakY <- da[year > 1961 & year < 2017] #ensure equal sample size
prebreakM <- dm[month <= 196210]
postbreakM <- dm[month > 196210 & month < (max(month)-201)] #ensure equal sample size

#subset according to climate results
preCBY <- da[year <= 1975 & year > 1930] #ensure equal sample size
postCBY <- da[year > 1975]
preCBM <- dm[month <= 197501 & month > 192902] #ensure equal sample size
postCBM <- dm[month > 197501]

#test for differences in means (annual)
{
  testmat1 <- matrix(nrow = 3, ncol=4)
  rownames(testmat1) <- c('De Bilt', 'Eelde', 'Maastricht')
  colnames(testmat1) <- c('t-Statistic', 'p-value', 'C.I. Lower', 'C.I. Upper')

  ttmAM <- t.test(postbreakY$maastricht, prebreakY$maastricht, paired = T)
  ttmAE <- t.test(postbreakY$eelde, prebreakY$eelde, paired = T)
  ttmAD <- t.test(postbreakY$de_bilt, prebreakY$de_bilt, paired = T)

  testmat1[1,1] <- ttmAD$statistic
  testmat1[2,1] <- ttmAE$statistic
  testmat1[3,1] <- ttmAM$statistic

  testmat1[1,2] <- ttmAD$p.value
  testmat1[2,2] <- ttmAE$p.value
  testmat1[3,2] <- ttmAM$p.value

  testmat1[1,3] <- ttmAD$conf.int[1:1]
  testmat1[2,3] <- ttmAE$conf.int[1:1]
  testmat1[3,3] <- ttmAM$conf.int[1:1]
```

```
  testmat1[1,4] <- ttmAD$conf.int[2:2]
  testmat1[2,4] <- ttmAE$conf.int[2:2]
  testmat1[3,4] <- ttmAM$conf.int[2:2]

  testmat21 <- matrix(nrow = 3, ncol=4)
  rownames(testmat21) <- c('De Bilt', 'Eelde', 'Maastricht')
  colnames(testmat21) <- c('t-Statistic', 'p-value', 'C.I. Lower', 'C.I. Upper')

  cbttmAM <- t.test(postCBY$maastricht, preCBY$maastricht, paired = T)
  cbttmAE <- t.test(postCBY$eelde, preCBY$eelde, paired = T)
  cbttmAD <- t.test(postCBY$de_bilt, preCBY$de_bilt, paired = T)

  testmat21[1,1] <- cbttmAD$statistic
  testmat21[2,1] <- cbttmAE$statistic
  testmat21[3,1] <- cbttmAM$statistic

  testmat21[1,2] <- cbttmAD$p.value
  testmat21[2,2] <- cbttmAE$p.value
  testmat21[3,2] <- cbttmAM$p.value

  testmat21[1,3] <- cbttmAD$conf.int[1:1]
  testmat21[2,3] <- cbttmAE$conf.int[1:1]
  testmat21[3,3] <- cbttmAM$conf.int[1:1]

  testmat21[1,4] <- cbttmAD$conf.int[2:2]
  testmat21[2,4] <- cbttmAE$conf.int[2:2]
  testmat21[3,4] <- cbttmAM$conf.int[2:2]
}

  #test for differences in means (monthly)
{
  testmat2 <- matrix(nrow = 3, ncol=4)
  rownames(testmat2) <- c('De Bilt', 'Eelde', 'Maastricht')
  colnames(testmat2) <- c('t-Statistic', 'p-value', 'C.I. Lower', 'C.I. Upper')

  ttmMM <- t.test(postbreakM$maastricht, prebreakM$maastricht, paired = T)
  ttmME <- t.test(postbreakM$eelde, prebreakM$eelde, paired = T)
  ttmMD <- t.test(postbreakM$de_bilt, prebreakM$de_bilt, paired = T)

  testmat2[1,1] <- ttmAD$statistic
  testmat2[2,1] <- ttmAE$statistic
  testmat2[3,1] <- ttmAM$statistic

  testmat2[1,2] <- ttmAD$p.value
  testmat2[2,2] <- ttmAE$p.value
  testmat2[3,2] <- ttmAM$p.value

  testmat2[1,3] <- ttmAD$conf.int[1:1]
```

```
  testmat2[2,3] <- ttmAE$conf.int[1:1]
  testmat2[3,3] <- ttmAM$conf.int[1:1]

  testmat2[1,4] <- ttmAD$conf.int[2:2]
  testmat2[2,4] <- ttmAE$conf.int[2:2]
  testmat2[3,4] <- ttmAM$conf.int[2:2]

  testmat23 <- matrix(nrow = 3, ncol=4)
  rownames(testmat23) <- c('De Bilt', 'Eelde', 'Maastricht')
  colnames(testmat23) <- c('t-Statistic', 'p-value', 'C.I. Lower', 'C.I. Upper')

  CBttmMM <- t.test(postCBM$maastricht, preCBM$maastricht, paired = T)
  CBttmME <- t.test(postCBM$eelde, preCBM$eelde, paired = T)
  CBttmMD <- t.test(postCBM$de_bilt, preCBM$de_bilt, paired = T)

  testmat23[1,1] <- CBttmMD$statistic
  testmat23[2,1] <- CBttmME$statistic
  testmat23[3,1] <- CBttmMM$statistic

  testmat23[1,2] <- CBttmMD$p.value
  testmat23[2,2] <- CBttmME$p.value
  testmat23[3,2] <- CBttmMM$p.value

  testmat23[1,3] <- CBttmMD$conf.int[1:1]
  testmat23[2,3] <- CBttmME$conf.int[1:1]
  testmat23[3,3] <- CBttmMM$conf.int[1:1]

  testmat23[1,4] <- CBttmMD$conf.int[2:2]
  testmat23[2,4] <- CBttmME$conf.int[2:2]
  testmat23[3,4] <- CBttmMM$conf.int[2:2]

}

  #use subsamples set earlier
{  #meanTable10y
  testmat3 <- matrix(nrow = 10, ncol=6)
  rownames(testmat3) <- 1:10
  colnames(testmat3) <- c('Base Year', 'Comparison Year', 't-Statistic', 'p-value', 'C.I. Lower',
  ↪  'C.I. Upper')

  j = 1
  for (i in 2:11) {
    testmat3[j,1] <- meanTable10y[1,Year]
    testmat3[j,2] <- meanTable10y[i,Year]
    testmat3[j,3] <- t.test(meanTable10y[1,2:4], meanTable10y[i,2:4])$statistic
    testmat3[j,4] <- t.test(meanTable10y[1,2:4], meanTable10y[i,2:4])$p.value
    testmat3[j,5] <- t.test(meanTable10y[1,2:4], meanTable10y[i,2:4])$conf.int[1:1]
    testmat3[j,6] <- t.test(meanTable10y[1,2:4], meanTable10y[i,2:4])$conf.int[2:2]
    j <- j + 1
```

```
  }

  #meanTable5y
  testmat4 <- matrix(nrow = 21, ncol=6)
  rownames(testmat4) <- 1:21
  colnames(testmat4) <- c('Base Year', 'Comparison Year', 't-Statistic', 'p-value', 'C.I. Lower', 'C.I. Upper')

  j = 1
  for (i in 2:22) {
    testmat4[j,1] <- meanTable5y[1,Year]
    testmat4[j,2] <- meanTable5y[i,Year]
    testmat4[j,3] <- t.test(meanTable5y[1,2:4], meanTable5y[i,2:4])$statistic
    testmat4[j,4] <- t.test(meanTable5y[1,2:4], meanTable5y[i,2:4])$p.value
    testmat4[j,5] <- t.test(meanTable5y[1,2:4], meanTable5y[i,2:4])$conf.int[1:1]
    testmat4[j,6] <- t.test(meanTable5y[1,2:4], meanTable5y[i,2:4])$conf.int[2:2]
    j <- j + 1
  }


  #medianTable10y
  testmat5 <- matrix(nrow = 10, ncol=6)
  rownames(testmat5) <- 1:10
  colnames(testmat5) <- c('Base Year', 'Comparison Year', 't-Statistic', 'p-value', 'C.I. Lower', 'C.I. Upper')

  j = 1
  for (i in 2:11) {
    testmat5[j,1] <- medianTable10y[1,Year]
    testmat5[j,2] <- medianTable10y[i,Year]
    testmat5[j,3] <- t.test(medianTable10y[1,2:4], medianTable10y[i,2:4])$statistic
    testmat5[j,4] <- t.test(medianTable10y[1,2:4], medianTable10y[i,2:4])$p.value
    testmat5[j,5] <- t.test(medianTable10y[1,2:4], medianTable10y[i,2:4])$conf.int[1:1]
    testmat5[j,6] <- t.test(medianTable10y[1,2:4], medianTable10y[i,2:4])$conf.int[2:2]
    j <- j + 1
  }

}

#test for homogeneity of variance (annual)
{
  testmat6 <- matrix(nrow = 3, ncol=5)
  rownames(testmat6) <- c('De Bilt', 'Eelde', 'Maastricht')
  colnames(testmat6) <- c('F-Statistic', 'p-value', 'C.I. Lower', 'C.I. Upper', 'Point Est.')

  FtmAM <- var.test(prebreakY$maastricht, postbreakY$maastricht)
  FtmAE <- var.test(prebreakY$eelde, postbreakY$eelde)
  FtmAD <- var.test(prebreakY$de_bilt, postbreakY$de_bilt)

```

```
  testmat6[1,1] <- FtmAD$statistic
  testmat6[2,1] <- FtmAE$statistic
  testmat6[3,1] <- FtmAM$statistic

  testmat6[1,5] <- FtmAD$estimate
  testmat6[2,5] <- FtmAE$estimate
  testmat6[3,5] <- FtmAM$estimate

  testmat6[1,2] <- FtmAD$p.value
  testmat6[2,2] <- FtmAE$p.value
  testmat6[3,2] <- FtmAM$p.value

  testmat6[1,3] <- FtmAD$conf.int[1:1]
  testmat6[2,3] <- FtmAE$conf.int[1:1]
  testmat6[3,3] <- FtmAM$conf.int[1:1]

  testmat6[1,4] <- FtmAD$conf.int[2:2]
  testmat6[2,4] <- FtmAE$conf.int[2:2]
  testmat6[3,4] <- FtmAM$conf.int[2:2]

  testmat22 <- matrix(nrow = 3, ncol=5)
  rownames(testmat22) <- c('De Bilt', 'Eelde', 'Maastricht')
  colnames(testmat22) <- c('F-Statistic', 'p-value', 'C.I. Lower', 'C.I. Upper', 'Point Est.')

  cbFtmAM <- var.test(preCBY$maastricht, postCBY$maastricht)
  cbFtmAE <- var.test(preCBY$eelde, postCBY$eelde)
  cbFtmAD <- var.test(preCBY$de_bilt, postCBY$de_bilt)

  testmat22[1,1] <- cbFtmAD$statistic
  testmat22[2,1] <- cbFtmAE$statistic
  testmat22[3,1] <- cbFtmAM$statistic

  testmat22[1,5] <- cbFtmAD$estimate
  testmat22[2,5] <- cbFtmAE$estimate
  testmat22[3,5] <- cbFtmAM$estimate

  testmat22[1,2] <- cbFtmAD$p.value
  testmat22[2,2] <- cbFtmAE$p.value
  testmat22[3,2] <- cbFtmAM$p.value

  testmat22[1,3] <- cbFtmAD$conf.int[1:1]
  testmat22[2,3] <- cbFtmAE$conf.int[1:1]
  testmat22[3,3] <- cbFtmAM$conf.int[1:1]

  testmat22[1,4] <- cbFtmAD$conf.int[2:2]
  testmat22[2,4] <- cbFtmAE$conf.int[2:2]
  testmat22[3,4] <- cbFtmAM$conf.int[2:2]

}
```

```r

#test for homogeneity of variance (monthly)
{
  testmat7 <- matrix(nrow = 3, ncol=5)
  rownames(testmat7) <- c('De Bilt', 'Eelde', 'Maastricht')
  colnames(testmat7) <- c('F-Statistic', 'p-value', 'C.I. Lower', 'C.I. Upper', 'Point Est.')

  FtmMM <- var.test(prebreakM$maastricht, postbreakM$maastricht)
  FtmME <- var.test(prebreakM$eelde, postbreakM$eelde)
  FtmMD <- var.test(prebreakM$de_bilt, postbreakM$de_bilt)

  testmat7[1,1] <- FtmMD$statistic
  testmat7[2,1] <- FtmME$statistic
  testmat7[3,1] <- FtmMM$statistic

  testmat7[1,5] <- FtmMD$estimate
  testmat7[2,5] <- FtmME$estimate
  testmat7[3,5] <- FtmMM$estimate

  testmat7[1,2] <- FtmMD$p.value
  testmat7[2,2] <- FtmME$p.value
  testmat7[3,2] <- FtmMM$p.value

  testmat7[1,3] <- FtmMD$conf.int[1:1]
  testmat7[2,3] <- FtmME$conf.int[1:1]
  testmat7[3,3] <- FtmMM$conf.int[1:1]

  testmat7[1,4] <- FtmMD$conf.int[2:2]
  testmat7[2,4] <- FtmME$conf.int[2:2]
  testmat7[3,4] <- FtmMM$conf.int[2:2]

  testmat24 <- matrix(nrow = 3, ncol=5)
  rownames(testmat24) <- c('De Bilt', 'Eelde', 'Maastricht')
  colnames(testmat24) <- c('F-Statistic', 'p-value', 'C.I. Lower', 'C.I. Upper', 'Point Est.')

  CBFtmMM <- var.test(preCBM$maastricht, postCBM$maastricht)
  CBFtmME <- var.test(preCBM$eelde, postCBM$eelde)
  CBFtmMD <- var.test(preCBM$de_bilt, postCBM$de_bilt)

  testmat24[1,1] <- CBFtmMD$statistic
  testmat24[2,1] <- CBFtmME$statistic
  testmat24[3,1] <- CBFtmMM$statistic

  testmat24[1,5] <- CBFtmMD$estimate
  testmat24[2,5] <- CBFtmME$estimate
  testmat24[3,5] <- CBFtmMM$estimate

  testmat24[1,2] <- CBFtmMD$p.value
  testmat24[2,2] <- CBFtmME$p.value
```

```r
  testmat24[3,2] <- CBFtmMM$p.value

  testmat24[1,3] <- CBFtmMD$conf.int[1:1]
  testmat24[2,3] <- CBFtmME$conf.int[1:1]
  testmat24[3,3] <- CBFtmMM$conf.int[1:1]

  testmat24[1,4] <- CBFtmMD$conf.int[2:2]
  testmat24[2,4] <- CBFtmME$conf.int[2:2]
  testmat24[3,4] <- CBFtmMM$conf.int[2:2]

}

#manual tests
{
  #manual Break test
  #right-tailed: H0: mean(diff <=0, H1: >0)
  #compute differences for pairs of obs
  diff <- postCBY[, .(de_bilt, eelde, maastricht)] - preCBY[, .(de_bilt, eelde, maastricht)]

  testmatMan <- matrix(nrow = 3, ncol=4)
  rownames(testmatMan) <- c('De Bilt', 'Eelde', 'Maastricht')
  colnames(testmatMan) <- c('t-Statistic', 'p-value', 'C.I. Lower', 'C.I. Upper')

  testD <- t.test(diff$de_bilt, alternative = "g", var.equal = F)
  testE <- t.test(diff$eelde, alternative = "g", var.equal = F)
  testM <- t.test(diff$maastricht, alternative = "g", var.equal = F)

  testmatMan[1,1] <- testD$statistic
  testmatMan[1,2] <- testD$p.value
  testmatMan[1,3] <- testD$conf.int[1:1]
  testmatMan[1,4] <- testD$conf.int[2:2]

  testmatMan[2,1] <- testE$statistic
  testmatMan[2,2] <- testE$p.value
  testmatMan[2,3] <- testE$conf.int[1:1]
  testmatMan[2,4] <- testE$conf.int[2:2]

  testmatMan[3,1] <- testM$statistic
  testmatMan[3,2] <- testM$p.value
  testmatMan[3,3] <- testM$conf.int[1:1]
  testmatMan[3,4] <- testM$conf.int[2:2]


}

#simple OLS
{
  OLS <- function(resp,pred){
      y <- as.matrix(resp)
```

```r
      X <- as.matrix(cbind(1,pred))
      beta <- solve(t(X)%*%X)%*%t(X)%*%y
      res <- as.matrix(y-beta[1]-beta[2]*X[,2])
      n <- length(resp)
      k <- ncol(X)
      VCV <- 1/(n-k)*as.numeric(t(res)%*%res)*solve(t(X)%*%X)
      se <- sqrt(diag(VCV))
      p_val <- rbind(2*pt(abs(beta[1]/se[1]),df=n-k,
                          lower.tail= FALSE),
                     2*pt(abs(beta[2]/se[2]),df=n-k,
                          lower.tail= FALSE))
      #bundle to return
      outMat <- matrix(nrow = 2, ncol=3)
      rownames(outMat) <- c('alpha', 'beta')
      colnames(outMat) <- c('estimate', 'se', 'p-value')
      outMat[1,1] <- beta[1:1]
      outMat[2,1] <- beta[2:2]
      outMat[1,2] <- se[1:1]
      outMat[2,2] <- se[2:2]
      outMat[1,3] <- p_val[1:1]
      outMat[2,3] <- p_val[2:2]
      return(outMat)

      OLS_res <- function(resp,pred){
        y <- as.matrix(resp)
        X <- as.matrix(cbind(1,pred))
        beta <- solve(t(X)%*%X)%*%t(X)%*%y
        res <- as.matrix(y-beta[1]-beta[2]*X[,2])

        return(res)
      }

  }

  #test correctness
  OLS(da$maastricht, da$year)

  lm(da$maastricht ~ da$year)

}

regMat <- OLS(da$maastricht, da$year)
regMat2 <- OLS(da$de_bilt, da$year)
regMat3 <- OLS(da$eelde, da$year)

regMatM <- OLS(dm$maastricht, dm$month)
regMatM2 <- OLS(dm$de_bilt, dm$month)
regMatM3 <- OLS(dm$eelde, dm$month)
```

```
#compute and store regressions for export to tables with stargazer (looks nicer)
{
  regYM <- lm(da$maastricht ~ da$year)
  regYD <- lm(da$de_bilt ~ da$year)
  regYE <- lm(da$eelde ~ da$year)

  regMM <- lm(dm$maastricht ~ dm$month)
  regMD <- lm(dm$de_bilt ~ dm$month)
  regME <- lm(dm$eelde ~ dm$month)

  regPreBYD <- lm(prebreakY$de_bilt ~ prebreakY$year)
  regPreBYE <- lm(prebreakY$eelde ~ prebreakY$year)
  regPreBYM <- lm(prebreakY$maastricht ~ prebreakY$year)

  regPostBYD <- lm(postbreakY$de_bilt ~ postbreakY$year)
  regPostBYE <- lm(postbreakY$eelde ~ postbreakY$year)
  regPostBYM <- lm(postbreakY$maastricht ~ postbreakY$year)

  regPreCBYD <- lm(preCBY$de_bilt ~ preCBY$year)
  regPreCBYE <- lm(preCBY$eelde ~ preCBY$year)
  regPreCBYM <- lm(preCBY$maastricht ~ preCBY$year)

  regPostCBYD <- lm(postCBY$de_bilt ~ postCBY$year)
  regPostCBYE <- lm(postCBY$eelde ~ postCBY$year)
  regPostCBYM <- lm(postCBY$maastricht ~ postCBY$year)

  regPreBMD <- lm(prebreakM$de_bilt ~ prebreakM$month)
  regPreBME <- lm(prebreakM$eelde ~ prebreakM$month)
  regPreBMM <- lm(prebreakM$maastricht ~ prebreakM$month)

  regPostBMD <- lm(postbreakM$de_bilt ~ postbreakM$month)
  regPostBME <- lm(postbreakM$eelde ~ postbreakM$month)
  regPostBMM <- lm(postbreakM$maastricht ~ postbreakM$month)

  regPreCBMD <- lm(preCBM$de_bilt ~ preCBM$month)
  regPreCBME <- lm(preCBM$eelde ~ preCBM$month)
  regPreCBMM <- lm(preCBM$maastricht ~ preCBM$month)

  regPostCBMD <- lm(postCBM$de_bilt ~ postCBM$month)
  regPostCBME <- lm(postCBM$eelde ~ postCBM$month)
  regPostCBMM <- lm(postCBM$maastricht ~ postCBM$month)

}

#white test for heteroscedasticity
{
  testmatHsced <- matrix(nrow = 6, ncol=2)
  rownames(testmatHsced) <- c('De Bilt, Yearly Data', 'Eelde, Yearly Data', 'Maastricht, Yearly
  ↪  Data',
```

```
                                'De Bilt, Monthly Data', 'Eelde, Monthly Data', 'Maastricht, Monthly Data')
  colnames(testmatHsced) <- c('Test Statistic', 'p-value')
  testmatHsced[1,1] <- white_lm(regYD)$statistic
  testmatHsced[4,1] <- white_lm(regMD)$statistic
  testmatHsced[1,2] <- white_lm(regYD)$p.value
  testmatHsced[4,2] <- white_lm(regMD)$p.value

  testmatHsced[2,1] <- white_lm(regYE)$statistic
  testmatHsced[5,1] <- white_lm(regME)$statistic
  testmatHsced[2,2] <- white_lm(regYE)$p.value
  testmatHsced[5,2] <- white_lm(regME)$p.value

  testmatHsced[3,1] <- white_lm(regYM)$statistic
  testmatHsced[6,1] <- white_lm(regMM)$statistic
  testmatHsced[3,2] <- white_lm(regYM)$p.value
  testmatHsced[6,2] <- white_lm(regMM)$p.value

  testmatHsced2 <- matrix(nrow = 6, ncol=2)
  rownames(testmatHsced2) <- c('De Bilt, before 1975', 'Eelde, before 1975', 'Maastricht, before 1975',
                                'De Bilt, after 1975', 'Eelde, after 1975', 'Maastricht, after 1975')
  colnames(testmatHsced2) <- c('Test Statistic', 'p-value')
  testmatHsced2[1,1] <- white_lm(regPreCBYD)$statistic
  testmatHsced2[4,1] <- white_lm(regPostCBMD)$statistic
  testmatHsced2[1,2] <- white_lm(regPreCBYD)$p.value
  testmatHsced2[4,2] <- white_lm(regPostCBMD)$p.value

  testmatHsced2[2,1] <- white_lm(regPreCBYE)$statistic
  testmatHsced2[5,1] <- white_lm(regPostCBME)$statistic
  testmatHsced2[2,2] <- white_lm(regPreCBYE)$p.value
  testmatHsced2[5,2] <- white_lm(regPostCBME)$p.value

  testmatHsced2[3,1] <- white_lm(regPreCBYM)$statistic
  testmatHsced2[6,1] <- white_lm(regPostCBMM)$statistic
  testmatHsced2[3,2] <- white_lm(regPreCBYM)$p.value
  testmatHsced2[6,2] <- white_lm(regPostCBMM)$p.value
}

########################CLEANUP AND EXPORT
{
  if (Sys.info()[7] == "ts") {

    ########################Do Plots & Tables###############################
    source("Tidy.R")
    source("Plots.R")
    source("Bootstrap.R")
    source("Tables.R")
```

```r
    ########################R File########################
    file.copy('TeamProject.R', '/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Code', overwrite = T)
    file.copy('Plots.R', '/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Code', overwrite = T)
    file.copy('Tables.R', '/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Code', overwrite = T)
    file.copy('Bootstrap.R', '/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Code', overwrite = T)
    file.copy('Tidy.R', '/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Code', overwrite = T)

    #credit OSS authors
    knitr::write_bib(c(.packages()),
                     "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/packages.bib")

    grateful::cite_packages(output = "paragraph", dependencies = T, include.RStudio = T,
                            out.dir = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/",
                            bib.file = "grateful.bib")
  }

}
```

2.4 Appendix D: Source Code (Bootstrap)

```r
####################BOOTSTRAP###
#BS functions
OLS_BS <- function(resp,pred){
  y <- as.matrix(resp)
  X <- as.matrix(cbind(1,pred))
  beta <- solve(t(X)%*%X)%*%t(X)%*%y
  res <- as.matrix(y-beta[1]-beta[2]*X[,2])
  n <- length(resp)
  I<- rep(1,n)
  k <- ncol(X)
  VCV <- 1/(n-k)*as.numeric(t(res)%*%res)*solve(t(X)%*%X)
  se <- sqrt(diag(VCV))
  p_val <- rbind(2*pt(abs(beta[1]/se[1]),df=n-k,
                      lower.tail= FALSE),
                 2*pt(abs(beta[2]/se[2]),df=n-k,
                      lower.tail= FALSE))
  v <- ((c(y)-beta[1:1]-beta[2:2]*(c(X)-pred[1]))^2)%*%t(I)
  var <- (1/n)*v[1] #MLE variance estimate
  #bundle to return
  outVec <- c(0,0,0)
  #rownames(outMat) <- c('alpha', 'beta')
  outVec[1] <- beta[1:1]
  outVec[2] <- beta[2:2]
  outVec[3] <- var
  return(outVec)
}
{
  S_2 <- function(y,x){
    n <- length(y)
    (1/(n-2))*OLS_BS(y,x)[3]
  }

  crit_val_BS <- function(Bootstrap){
    alpha <- 0.05 Choose a significance level alpha
    c.alpha.star <- quantile(Bootstrap, probs = 1 - alpha)    # Get the bootstrap critical value
    return(c.alpha.star)
  }

  Q_BS <- function(Beta, Beta.0, X, Y){
    S_xx <- var(X)
    S_2 <- cov(X, Y)
    Q_BS <- (Beta - Beta.0)/sqrt(S_2/S_xx)
    return(Q_BS)
  }
```

```
ResidualVector<- function(n,X,Y,a,b){
  resid<-rep(0,n)
  for(i in 1:length(Y)){
    resid[i]= Y[i]-a-b*X[i]
  }
  return(resid)
}

BS_Int <- function(b,crit_alpha_half,crit_one_minus_alpha_half, S_2, X){

  lb <- b-(crit_alpha_half*sqrt(S_2/var(X)))
  ub <- b-(crit_one_minus_alpha_half*sqrt(S_2/var(X)))

  BS_CI <- c(0,0)
  BS_CI[1] <- lb
  BS_CI[2] <- ub

  return(BS_CI)
}

Resid_BS <- function(n, X, Y, resid, a, beta){
  B<- 9999
  Q.Star<- rep(NA,B)
  BetaLSstar <- rep(NA,B)
  AlphaLSstar <- rep(NA,B)

  for(b in 1:B){

    J <- sample.int(length(Y), size = n, replace= TRUE)
    resid.star <- resid[J]
    X.star <- X #fix
    Y.star <- a +beta*X.star + resid.star #We take alpha* and beta* as estimators in order to
    ↪  bootstrap the residuals, we can also take alpha*=alpha.0 and beta*=beta.0

    X.star.bar<- mean(X.star)
    Y.star.bar<- mean(Y.star)
    S.XX.star <- var(X.star)
    S.XY.star <- cov(X.star, Y.star)

    BetaLSstar[b]<-S.XY.star/S.XX.star
    AlphaLSstar[b]<-Y.star.bar-(BetaLSstar[b]*X.star.bar)

    S.squared.star<-(1/(n-2))*sum(n, (Y.star-AlphaLSstar[b]-BetaLSstar[b]*X.star)^2)

    Q.Star[b]<- (BetaLSstar[b]-beta)/sqrt(S.squared.star/S.XX.star)

  }
  return(Q.Star)
```

```r
}

Pairs_BS <- function(n,X,Y,BetaLS){ #pages 33-34 BS notes

  B <- 9999 #numreps
  Q.star <- rep(NA, B) #ret vector
  BetaLSstar <- rep(NA, B) #beta
  AlphaLSstar <- rep(NA, B) #alpha

  for(b in 1:B){

    J <- sample.int(nrow(da), size = n, replace= TRUE)
    X.star <- X[J]
    Y.star <- Y[J]

    X.star.bar <- mean(X.star)
    Y.star.bar <- mean(Y.star)
    S.XX.star <- var(X.star)
    S.XY.star <- cov(X.star, Y.star)

    BetaLSstar[b] <- S.XY.star/S.XX.star  #BS beta
    AlphaLSstar[b] <- Y.star.bar- BetaLSstar[b]*X.star.bar #BS alpha

    S.squared.star <- (1/(n-2))*sum(n, (Y.star-AlphaLSstar[b]-BetaLSstar[b]*X.star)^2)

    Q.star[b] <- (BetaLSstar[b]-BetaLS)/sqrt(S.squared.star/S.XX.star) #bootstrap Q
  }
  return(Q.star)
}

BS_t <- function(X){ #H0: Q*<=0 H1: Q*>0
  B<-9999
  Q.star <- rep(NA,B)
  n<- length(X)
  t.n <- t.test(X, alternative = "greater", mu = 0)$statistic
  for(b in 1:B){

    J <- sample.int(n,size = n, replace = TRUE)
    X_star <- X[J]
    X_bar_star <- mean(X_star)
    X_Sd_bar <- sd(X_star)

    Q.star[b]<- sqrt(n)*(X_bar_star-mean(X))/X_Sd_bar
    p.val <- sum(Q.star > t.n) #see p.27 of bootstrap pdf
  }

  return(list( Q = Q.star, p = p.val, t = t.n))
}
```

```
  BS_CI_t <- function(X, critical_alpha_half, critical_1min__alpha_half){

    lb <- mean(X) -(critical_alpha_half*sd(X))/sqrt(length(X))
    ub <- mean(X)-(critical_1min__alpha_half*sd(X))/sqrt(length(X))

    CI <- c(0,0)
    CI[1]<-lb
    CI[2]<-ub

    return(CI)
  }
}

#prep data
{
  aD <- OLS_BS(da$de_bilt, da$year)[1]
  bD <- OLS_BS(da$de_bilt, da$year)[2]
  aM <- OLS_BS(da$maastricht, da$year)[1]
  bM <- OLS_BS(da$maastricht, da$year)[2]
  aE <- OLS_BS(da$eelde, da$year)[1]
  bE <- OLS_BS(da$eelde, da$year)[2]

  aDM <- OLS_BS(dm$de_bilt, dm$month)[1]
  bDM <- OLS_BS(dm$de_bilt, dm$month)[2]
  aMM <- OLS_BS(dm$maastricht, dm$month)[1]
  bMM <- OLS_BS(dm$maastricht, dm$month)[2]
  aEM <- OLS_BS(dm$eelde, dm$month)[1]
  bEM <- OLS_BS(dm$eelde, dm$month)[2]
}

#BS analysis
{
  Beta.0 <- 0 #beta under H0: Beta.0<=0 H1: Beta.0>0
  set.seed(234987)

  n <- nrow(da)
  alpha <- 0.05

  #get results
  {
    #Quantities
    QMY <- Q_BS(bM, Beta.0, da$year, da$maastricht)
    QDY <- Q_BS(bM, Beta.0, da$year, da$de_bilt)
    QEY <- Q_BS(bM, Beta.0, da$year, da$eelde)

    QMM <- Q_BS(bM, Beta.0, dm$maastricht, dm$maastricht)
    QDM <- Q_BS(bM, Beta.0, dm$maastricht, dm$de_bilt)
    QEM <- Q_BS(bM, Beta.0, dm$maastricht, dm$eelde)
```

```r

#BS
PairsBSM <- Pairs_BS(n, da$year, da$maastricht, bM)
PairsBSD <- Pairs_BS(n, da$year, da$de_bilt, bD)
PairsBSE <- Pairs_BS(n, da$year, da$eelde, bE)
#monthly
PairsBSMM <- Pairs_BS(n, dm$month, dm$maastricht, bMM)
PairsBSDM <- Pairs_BS(n, dm$month, dm$de_bilt, bDM)
PairsBSEM <- Pairs_BS(n, dm$month, dm$eelde, bEM)

ResidBSM <- Resid_BS(length(da$year), da$maastricht, da$year, ResidualVector(length(da$year), da$maastricht, da$year, aM, bM), aM, bM)
ResidBSD <- Resid_BS(length(da$year), da$de_bilt, da$year, ResidualVector(length(da$year), da$de_bilt, da$year, aD, bD), aD, bD)
ResidBSE <- Resid_BS(length(da$year), da$eelde, da$year, ResidualVector(length(da$year), da$eelde, da$year, aE, bE), aE, bE)

#monthly
ResidBSMM <- Resid_BS(length(dm$month), dm$maastricht, dm$month, ResidualVector(length(dm$month), dm$maastricht, dm$mont, aMM, bMM), aMM, bMM)
ResidBSDM <- Resid_BS(length(dm$month), dm$de_bilt, dm$month, ResidualVector(length(dm$month), dm$de_bilt, dm$mont, aDM, bDM), aDM, bDM)
ResidBSEM <- Resid_BS(length(dm$month), dm$eelde, dm$month, ResidualVector(length(dm$month), dm$eelde, dm$mont, aEM, bEM), aEM, bEM)

#CIs
CIPairsM <- BS_Int(bM, quantile(PairsBSM, probs = 1-(alpha/2)), quantile(PairsBSM, probs = (alpha/2)), S_2(da$year, da$maastricht), da$maastricht)
CIResidM <- BS_Int(bM, quantile(ResidBSM, probs = 1-(alpha/2)), quantile(ResidBSM, probs = (alpha/2)), S_2(da$year, da$maastricht), da$maastricht)

CIPairsD <- BS_Int(bD, quantile(PairsBSD, probs = 1-(alpha/2)), quantile(PairsBSD, probs = (alpha/2)), S_2(da$year, da$de_bilt), da$de_bilt)
CIResidD <- BS_Int(bD, quantile(ResidBSD, probs = 1-(alpha/2)), quantile(ResidBSD, probs = (alpha/2)), S_2(da$year, da$de_bilt), da$de_bilt)

CIPairsE <- BS_Int(bE, quantile(PairsBSE, probs = 1-(alpha/2)), quantile(PairsBSE, probs = (alpha/2)), S_2(da$year, da$eelde), da$eelde)
CIResidE <- BS_Int(bE, quantile(ResidBSE, probs = 1-(alpha/2)), quantile(ResidBSE, probs = (alpha/2)), S_2(da$year, da$eelde), da$eelde)

#monthly
CIPairsMM <- BS_Int(bMM, quantile(PairsBSMM, probs = 1-(alpha/2)), quantile(PairsBSMM, probs = (alpha/2)), S_2(dm$month, dm$maastricht), dm$maastricht)
CIResidMM <- BS_Int(bMM, quantile(ResidBSMM, probs = 1-(alpha/2)), quantile(ResidBSMM, probs = (alpha/2)), S_2(dm$month, dm$maastricht), dm$maastricht)

CIPairsDM <- BS_Int(bDM, quantile(PairsBSDM, probs = 1-(alpha/2)), quantile(PairsBSDM, probs = (alpha/2)), S_2(dm$month, dm$de_bilt), dm$de_bilt)
```

```
  CIResidDM <- BS_Int(bDM, quantile(ResidBSDM, probs = 1-(alpha/2)), quantile(ResidBSDM, probs = (alpha/2)), S_2(dm$month, dm$de_bilt), dm$de_bilt)

  CIPairsEM <- BS_Int(bEM, quantile(PairsBSEM, probs = 1-(alpha/2)), quantile(PairsBSEM, probs = (alpha/2)), S_2(dm$month, dm$eelde), dm$eelde)
  CIResidEM <- BS_Int(bEM, quantile(ResidBSEM, probs = 1-(alpha/2)), quantile(ResidBSEM, probs = (alpha/2)), S_2(dm$month, dm$eelde), dm$eelde)

}

#export results
{
  BSmat1 <- matrix(nrow = 12, ncol=3)
  rownames(BSmat1) <- c('De Bilt, Yearly Data, Pairs', 'De Bilt, Yearly Data, Residuals', 'Eelde, Yearly Data, Pairs',
                        'Eelde, Yearly Data, Residuals','Maastricht, Yearly Data, Pairs','Maastricht, Yearly Data, Residuals',
                        'De Bilt, Monthly Data, Pairs', 'De Bilt, Monthly Data, Residuals', 'Eelde, Monthly Data, Pairs',
                        'Eelde, Monthly Data, Residuals','Maastricht, Monthly Data, Pairs','Maastricht, Monthly Data, Residuals')
  colnames(BSmat1) <- c('$Q^*$', 'CI lower', 'CI upper')

  BSmat1[1,1] <- QDY
  BSmat1[3,1] <- QEY
  BSmat1[5,1] <- QMY
  BSmat1[7,1] <- QDY
  BSmat1[9,1] <- QEY
  BSmat1[11,1] <- QMY


  BSmat1[1,2] <- CIPairsD[1:1]
  BSmat1[2,2] <- CIResidD[1:1]

  BSmat1[3,2] <- CIPairsE[1:1]
  BSmat1[4,2] <- CIResidE[1:1]

  BSmat1[5,2] <- CIPairsM[1:1]
  BSmat1[6,2] <- CIResidM[1:1]

  BSmat1[7,2] <- CIPairsDM[1:1]
  BSmat1[8,2] <- CIResidDM[1:1]

  BSmat1[9,2] <- CIPairsEM[1:1]
  BSmat1[10,2] <- CIResidEM[1:1]

  BSmat1[11,2] <- CIPairsMM[1:1]
  BSmat1[12,2] <- CIResidMM[1:1]

```

```
    BSmat1[1,3] <- CIPairsD[2:2]
    BSmat1[2,3] <- CIResidD[2:2]

    BSmat1[3,3] <- CIPairsE[2:2]
    BSmat1[4,3] <- CIResidM[2:2]

    BSmat1[5,3] <- CIPairsM[2:2]
    BSmat1[6,3] <- CIResidM[2:2]

    BSmat1[7,3] <- CIPairsDM[2:2]
    BSmat1[8,3] <- CIResidDM[2:2]

    BSmat1[9,3] <- CIPairsEM[2:2]
    BSmat1[10,3] <- CIResidMM[2:2]

    BSmat1[11,3] <- CIPairsMM[2:2]
    BSmat1[12,3] <- CIResidMM[2:2]
  }

  ##BS t-test for Climate Break
  {
    #diff <- postCBY[, .(de_bilt, eelde, maastricht)] - preCBY[, .(de_bilt, eelde, maastricht)]
    ↪  (already exists)
    diffM <- postCBM[, .(de_bilt, eelde, maastricht)] - preCBM[, .(de_bilt, eelde, maastricht)]

    BStM <- BS_t(diff$maastricht)
    BSCItM <- BS_CI_t(diff$maastricht, quantile(BStM$Q,probs=1-(alpha/2)),
    ↪  quantile(BStM$Q,probs=(alpha/2)))

    BStD <- BS_t(diff$de_bilt)
    BSCItD <- BS_CI_t(diff$de_bilt, quantile(BStD$Q,probs=1-(alpha/2)),
    ↪  quantile(BStD$Q,probs=(alpha/2)))

    BStE <- BS_t(diff$eelde)
    BSCItE <- BS_CI_t(diff$eelde, quantile(BStE$Q,probs=1-(alpha/2)),
    ↪  quantile(BStE$Q,probs=(alpha/2)))

    #monthly
    BStMM <- BS_t(diffM$maastricht)
    BSCItMM <- BS_CI_t(diffM$maastricht, quantile(BStMM$Q,probs=1-(alpha/2)),
    ↪  quantile(BStMM$Q,probs=(alpha/2)))

    BStDM <- BS_t(diffM$de_bilt)
    BSCItDM <- BS_CI_t(diffM$de_bilt, quantile(BStDM$Q,probs=1-(alpha/2)),
    ↪  quantile(BStDM$Q,probs=(alpha/2)))

    BStEM <- BS_t(diffM$eelde)
    BSCItEM <- BS_CI_t(diffM$eelde, quantile(BStEM$Q,probs=1-(alpha/2)),
    ↪  quantile(BStEM$Q,probs=(alpha/2)))
```

```
  }


  #export results
  {
    BSmat2 <- matrix(nrow = 6, ncol=4)
    rownames(BSmat2) <- c('De Bilt, Yearly Data', 'Eelde, Yearly Data','Maastricht, Yearly Data',
                          'De Bilt, Monthly Data', 'Eelde, Monthly Data','Maastricht, Monthly
                          ↪  Data')
    colnames(BSmat2) <- c('$t_n$', 'p-value', 'CI lower', 'CI upper')

    BSmat2[1,1] <- BStD$t
    BSmat2[2,1] <- BStE$t
    BSmat2[3,1] <- BStM$t

    BSmat2[1,2] <- BStD$p
    BSmat2[2,2] <- BStE$p
    BSmat2[3,2] <- BStM$p

    BSmat2[4,1] <- BStDM$t
    BSmat2[5,1] <- BStEM$t
    BSmat2[6,1] <- BStMM$t

    BSmat2[4,2] <- BStDM$p
    BSmat2[5,2] <- BStEM$p
    BSmat2[6,2] <- BStMM$p

    BSmat2[1,3] <- BSCItD[1]
    BSmat2[2,3] <- BSCItE[1]
    BSmat2[3,3] <- BSCItM[1]

    BSmat2[1,4] <- BSCItD[2]
    BSmat2[2,4] <- BSCItE[2]
    BSmat2[3,4] <- BSCItM[2]

    BSmat2[4,3] <- BSCItDM[1]
    BSmat2[5,3] <- BSCItEM[1]
    BSmat2[6,3] <- BSCItMM[1]

    BSmat2[4,4] <- BSCItDM[2]
    BSmat2[5,4] <- BSCItEM[2]
    BSmat2[6,4] <- BSCItMM[2]
  }

}
```

2.5 Appendix E: Source Code (Data Tidying)

```r
###############################tidy#############################

daC <- da[, .(Maastricht = maastricht, Eelde = eelde, De.Bilt = de_bilt, Year = year)]
daLong <- melt(daC, id.vars = c("Year"), measure.vars = c("Maastricht", "Eelde", "De.Bilt"),
               variable.factor = T, variable.name = "City", value.name = "Temperature")
citymeanA <- daLong[, Citymean := mean(Temperature), by = City]

dmsC <- dms[, .(Maastricht = maastricht, Eelde = eelde, De.Bilt = de_bilt, Month = month )]
dmsLong <- melt(dmsC, id.vars = c("Month"), measure.vars = c("Maastricht", "Eelde", "De.Bilt"),
                variable.factor = T, variable.name = "City", value.name = "Temperature")
citymeanMS <- dmsLong[, Citymean := mean(Temperature), by = City]

dmC <- dm[, .(Maastricht = maastricht, Eelde = eelde, De.Bilt = de_bilt, Month = month )]
dmLong <- melt(dmC, id.vars = c("Month"), measure.vars = c("Maastricht", "Eelde", "De.Bilt"),
               variable.factor = T, variable.name = "City", value.name = "Temperature")
citymeanM <- dmLong[, Citymean := mean(Temperature), by = City]

dmBackupC <- dmbackup[, .(Maastricht = maastricht, Eelde = eelde, De.Bilt = de_bilt, Month = month
↪ )]
dmLongBackup <- melt(dmBackupC, id.vars = c("Month"), measure.vars = c("Maastricht", "Eelde",
↪ "De.Bilt"),
               variable.factor = T, variable.name = "City", value.name = "Temperature")
citymeanMBackup <- dmLong[, Citymean := mean(Temperature), by = City]

ddC <- dd[, .(Maastricht = maastricht, Eelde = eelde, De.Bilt = de_bilt, Date = date )]
ddLong <- melt(ddC, id.vars = c("Date"), measure.vars = c("Maastricht", "Eelde", "De.Bilt"),
               variable.factor = T, variable.name = "City", value.name = "Temperature")
citymeanD <- ddLong[, Citymean := mean(Temperature), by = City]


rolling10_5L <- melt(rollingMean10_5, id.vars = c("Year"), measure.vars = c("Maastricht", "Eelde",
↪ "De.Bilt"),
                     variable.factor = T, variable.name = "City", value.name = "Temperature")

rolling20_10L <- melt(rollingMean20_10, id.vars = c("Year"), measure.vars = c("Maastricht",
↪ "Eelde", "De.Bilt"),
                      variable.factor = T, variable.name = "City", value.name = "Temperature")
```

2.6 Appendix F: Source Code (Plots)

```r
##gert required subsets
#4-way plot subsets
march <- subsetMonthLong(3)
june <- subsetMonthLong(6)
september <- subsetMonthLong(9)
december <- subsetMonthLong(12)

#rolling window plots
rollingMean10_5 <- xYearYoverlapStat(10, 5, mean)
rollingMean20_10 <- xYearYoverlapStat(20, 10, mean)

#date plot
april7 <- subsetDateLong(407)
#####################Plots######################
#raw data
TSA <- ggplot(daLong, aes(x = Year, y = Temperature)) +
  geom_line(aes(color = City)) + labs(y = 'Temperature', x = 'Year') +
  theme_minimal() + scale_color_tableau() +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5)) +
  xlim(1905, 2025) + ggtitle("Annual Data")

ggsave("TSA.png",  bg = "white", dpi = "retina", width = 20, height = 10, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Raw")

TSABP <- ggplot(daLong, aes(x = Year, y = Temperature)) +
  geom_line(aes(color = City)) + labs(y = 'Temperature', x = 'Year') +
  theme_minimal() + scale_color_tableau() + geom_vline(xintercept = 1961, linetype = 'dashed',
  ↪  colour = '#76B7B2') +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5)) +
  xlim(1945, 1975) + ggtitle("Annual Data: Breakpoint")

ggsave("TSA_BP.png",  bg = "white", dpi = "retina", width = 20, height = 10, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Raw")

TS4_7 <- ggplot(april7, aes(x = Date, y = Temperature, xmin = as.Date("1907-01-01", "%Y-%m-%d"),
↪  xmax = as.Date("2023-01-01", "%Y-%m-%d"))) +
  geom_line(aes(color = City)) + labs(y = 'Temperature', x = 'Year') +
  theme_minimal() + scale_color_tableau() +
  theme( panel.grid.minor = element_blank(), panel.grid.major = element_blank(), plot.title =
  ↪  element_text(hjust = 0.5), legend.position = "bottom") +
  ggtitle("Temperatures on April 7") +
  geom_text(aes(as.Date("2022-04-07"), 11), label=10, colour = '#4E79A7') + #maastricht
  geom_text(aes(as.Date("2022-04-07"), 9), label = 9,  colour = '#F28E2B') + #eelde
  geom_text(aes(as.Date("2022-04-07"), 10), label = 9,  colour = '#E15759')  #debilt

ggsave("TS4_7.png",  bg = "white", dpi = "retina", width = 20, height = 12, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Raw")
```

```r

#red E15759
#blue 4E79A7
#orange F28E2B

# dat <- dmLong[, Month := yearmon(Month)]
TSM <- ggplot(dmLong, aes(x = Month, y = Temperature)) +
  geom_line(aes(color = City)) + labs(y = 'Temperature', x = 'Year') +
  theme_minimal() + scale_color_tableau() +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5)) +
  xlim(190501, 202501) + ggtitle("Monthly Data")

ggsave("TSM.png",  bg = "white", dpi = "retina", width = 20, height = 10, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Raw")

TSMbackup <- ggplot(dmLongBackup, aes(x = Month, y = Temperature)) +
  geom_line(aes(color = City)) + labs(y = 'Temperature', x = 'Year') +
  theme_minimal() + scale_color_tableau() +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5)) +
  xlim(190501, 202501) + ggtitle("Monthly Data April - November")

ggsave("TSMBackup.png",  bg = "white", dpi = "retina", width = 20, height = 10, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Raw")

TSMS <- ggplot(dmsLong, aes(x = Month, y = Temperature)) +
  geom_line(aes(color = City)) + labs(y = 'Temperature', x = 'Year') +
  theme_minimal() + scale_color_tableau() +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5)) +
  xlim(190501, 202501) + ggtitle("Smoothed Monthly Data")

ggsave("TSMS.png",  bg = "white", dpi = "retina", width = 20, height = 10, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Raw")


#normality
qqY <-  ggplot(daLong, aes(sample = Temperature)) +
  stat_qq(aes(color = City)) +
  labs(y = "Weight", x = "Theoretical") +theme_minimal() + ggtitle("Annual Data Q-Q Plot") +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5)) +
  scale_color_tableau()


ggsave("qqY.png",  bg = "white", dpi = "retina", width = 20, height = 10, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Raw")

qqM <-  ggplot(dmLong, aes(sample = Temperature)) +
  stat_qq(aes(color = City)) +
  labs(y = "Weight", x = "Theoretical") +theme_minimal() + ggtitle("Monthly Data Q-Q Plot") +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5)) +
```

```
  scale_color_tableau()


ggsave("qqM.png",  bg = "white", dpi = "retina", width = 20, height = 10, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Raw")

qqMs <-  ggplot(dmsLong, aes(sample = Temperature)) +
  stat_qq(aes(color = City)) +
  labs(y = "Weight", x = "Theoretical") +theme_minimal() + ggtitle("Smoothed Monthly Data Q-Q
  ↪  Plot") +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5)) +
  scale_color_tableau()


ggsave("qqMs.png",  bg = "white", dpi = "retina", width = 20, height = 10, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Raw")

qqD <-  ggplot(ddLong, aes(sample = Temperature)) +
  stat_qq(aes(color = City)) +
  labs(y = "Weight", x = "Theoretical") +theme_minimal() + ggtitle("Daily Data Q-Q Plot") +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5)) +
  scale_color_tableau()


ggsave("qqD.png",  bg = "white", dpi = "retina", width = 20, height = 10, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Raw")

#histograms
histY <- ggplot(daLong, aes(x = Temperature, color = City)) +
  geom_histogram(fill="white", alpha=0.5, position="dodge", bins=45) +
  geom_vline(data=citymeanA, aes(xintercept = Citymean, color = City), linetype = "dashed") +
  theme_minimal() + ylab("Density") + ggtitle("Annual Mean Temperatures") +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5)) +
  scale_color_tableau()

ggsave("Ahist.png",  bg = "white", dpi = "retina", width = 20, height = 10, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Raw")


histM <- ggplot(dmLong, aes(x = Temperature, color = City)) +
  geom_histogram(fill="white", alpha=0.5, position="dodge", bins=45) +
  geom_vline(data=citymeanM, aes(xintercept = Citymean, color = City), linetype = "dashed") +
  theme_minimal() + ylab("Density") + ggtitle("Monthly Mean Temperatures") +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5)) +
  scale_color_tableau()

ggsave("Mhist.png",  bg = "white", dpi = "retina", width = 20, height = 10, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Raw")

```

```r

#densities

densplotyears <- ggplot(daLong, aes(x = Temperature, color = City)) + geom_density() +
  geom_vline(data=citymeanA, aes(xintercept = Citymean, color = City), linetype = "dashed") +
  theme_minimal() + ylab("Density") + ggtitle("Annual Mean Temperatures") +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5)) +
  scale_color_tableau()


ggsave("AD.png",  bg = "white", dpi = "retina", width = 20, height = 10, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Raw")

densplotmonths <- ggplot(dmLong, aes(x = Temperature, color = City)) + geom_density() +
  geom_vline(data=citymeanM, aes(xintercept = Citymean, color = City), linetype = "dashed") +
  theme_minimal() + ylab("Density")+ ggtitle("Monthly Mean Temperatures") +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5)) +
  scale_color_tableau()

ggsave("MD.png",  bg = "white", dpi = "retina", width = 20, height = 10, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Raw")

densplotmonthsBackup <- ggplot(dmLongBackup, aes(x = Temperature, color = City)) + geom_density()
↪ +
  geom_vline(data=citymeanMBackup, aes(xintercept = Citymean, color = City), linetype = "dashed")
  ↪ +
  theme_minimal() + ylab("Density")+ ggtitle("Monthly Mean Temperatures") +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5)) +
  scale_color_tableau()

ggsave("MDBackup.png",  bg = "white", dpi = "retina", width = 20, height = 10, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Raw")


densplotmonthsS <- ggplot(dmsLong, aes(x = Temperature, color = City)) + geom_density() +
  geom_vline(data=citymeanMS, aes(xintercept = Citymean, color = City), linetype = "dashed") +
  theme_minimal() + ylab("Density")+ ggtitle("Smoothed Monthly Mean Temperatures") +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5)) +
  scale_color_tableau()

ggsave("MSD.png",  bg = "white", dpi = "retina", width = 20, height = 10, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Raw")

densplotDays <- ggplot(ddLong, aes(x = Temperature, color = City)) + geom_density() +
  geom_vline(data=citymeanD, aes(xintercept = Citymean, color = City), linetype = "dashed") +
  theme_minimal() + ylab("Density")+ ggtitle("Daily Mean Temperatures") +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5)) +
  scale_color_tableau()

```

```r
ggsave("DD.png",  bg = "white", dpi = "retina", width = 20, height = 10, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Raw")

densplotRoll10_5 <- ggplot(rolling10_5L, aes(x = Temperature, color = City))+ geom_density() +
  theme_minimal() + ylab("Density")+ ggtitle("Rolling 5 Year Window 10 Year Mean Temperatures") +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5))+
  scale_color_tableau()

ggsave("10_5D.png",  bg = "white", dpi = "retina", width = 20, height = 10, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Raw")

densplotRoll20_10 <- ggplot(rolling20_10L, aes(x = Temperature, color = City))+ geom_density() +
  theme_minimal() + ylab("Density")+ ggtitle("Rolling 10 Year Window 20 Year Mean Temperatures") +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5))+
  scale_color_tableau()

ggsave("20_10D.png",  bg = "white", dpi = "retina", width = 20, height = 10, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Raw")


marchD2 <- ggplot(march, aes(x = Temperature, color = City)) + geom_density() +
  geom_vline(aes(xintercept = Citymean, color = City), linetype = "dashed") +
  theme_minimal() + ylab("Density")+ ggtitle("March") +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5, size = 12),
  ↪  legend.position="none") +
  scale_color_tableau()

juneD2 <- ggplot(june, aes(x = Temperature, color = City)) + geom_density() +
  geom_vline(aes(xintercept = Citymean, color = City), linetype = "dashed") +
  theme_minimal() + ylab("Density")+ ggtitle("June") +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5, size = 12),
  ↪  legend.position="none") +
  scale_color_tableau()

septemberD2 <- ggplot(september, aes(x = Temperature, color = City)) + geom_density() +
  geom_vline(aes(xintercept = Citymean, color = City), linetype = "dashed") +
  theme_minimal() + ylab("Density")+ ggtitle("September") +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5, size = 12),
  ↪  legend.position="none") +
  scale_color_tableau()

decemberD2 <- ggplot(december, aes(x = Temperature, color = City)) + geom_density() +
  geom_vline(aes(xintercept = Citymean, color = City), linetype = "dashed") +
  theme_minimal() + ylab("Density")+ ggtitle("December") +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5, size = 12),
  ↪  legend.position="none") +
  scale_color_tableau()

fourwayplot <- marchD2 + juneD2 + septemberD2 + decemberD2 +
```

```r
  plot_layout(guides = "collect") & theme(legend.position = "bottom")

fourwayplot <-  fourwayplot +  plot_annotation(title = 'Mean Temperatures in Different Months',
                                               caption = 'Means computed individually',
                                               theme = theme(plot.title = element_text(hjust = 0.5))
)
#, family="Times New Roman"

ggsave("4wayD.png",  bg = "white", dpi = "retina", width = 20, height = 15, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Raw")


#test stats for 5y mean tables
#graph building blocks

hyptestdatw <- as.data.table(cbind(seq(from=1916, to=2016, by = 5), testmat4[,4]))
colnames(hyptestdatw) <- c("Year", "p-values")

hyptestdat <- melt(hyptestdatw, id.vars = c("Year"), measure.vars = "p-values", variable.factor = F)

hyptestplot <- ggplot(hyptestdat, aes(x= hyptestdat$Year, y= hyptestdat$value)) +
  geom_line(colour = '#E15759') +
  geom_hline(yintercept = 0.05, colour = '#4E79A7') +
  geom_text(aes(2000, 0.05, label = 0.05, vjust = -0.5, colour = '#4E79A7')) +
  geom_hline(yintercept = 0.1, colour = '#F28E2B') +
  geom_text(aes(2000, 0.1, label = 0.1, vjust = -0.5, colour = '#F28E2B')) +
  labs(y = 'p-values', x = 'Year') +
  theme_minimal() + scale_color_tableau() +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5), legend.position = "none") +
  ggtitle("t-test Significance Levels, t-tests on 5-year Means, Base Year 1911")

ggsave("hyptestplot.png",  bg = "white", dpi = "retina", width = 20, height = 15, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures")

##regression plots
#red E15759
#blue 4E79A7
#orange F28E2B
regPlotAllY <- ggplot(daLong, aes(x = Year, y = Temperature)) +
  geom_point(aes(colour = City)) + labs(y = 'Temperature', x = 'Year') +
  theme_minimal() + scale_color_tableau() +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5)) +
  xlim(1905, 2025) + ggtitle("Yearly Data")

ggsave("regYD.png",  bg = "white", dpi = "retina", width = 20, height = 15, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Regs")
```

```r
regYE <- ggplot(filter(daLong, City == 'Eelde'), aes(x = Year, y = Temperature)) +
  geom_point(colour = '#F28E2B') + labs(y = 'Temperature', x = 'Year') +
  theme_minimal() + scale_color_tableau() +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5)) +
  xlim(1905, 2025) + ggtitle("Regression of Year on Temperature in Eelde") +
  geom_smooth(method='lm', color = '#499894', fill = '#86BCB6')

ggsave("regYE.png",  bg = "white", dpi = "retina", width = 20, height = 15, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Regs")

regYD <- ggplot(filter(daLong, City == 'De.Bilt'), aes(x = Year, y = Temperature)) +
  geom_point(colour = '#E15759') + labs(y = 'Temperature', x = 'Year') +
  theme_minimal() + scale_color_tableau() +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5)) +
  xlim(1905, 2025) + ggtitle("Regression of Year on Temperature in De Bilt") +
  geom_smooth(method='lm', color = '#499894', fill = '#86BCB6')

ggsave("regYD.png",  bg = "white", dpi = "retina", width = 20, height = 15, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Regs")

regYM <- ggplot(filter(daLong, City == 'Maastricht'), aes(x = Year, y = Temperature)) +
  geom_point(colour = '#4E79A7') + labs(y = 'Temperature', x = 'Year') +
  theme_minimal() + scale_color_tableau() +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5)) +
  xlim(1905, 2025) + ggtitle("Regression of Year on Temperature in Maastricht") +
  geom_smooth(method='lm', color = '#499894', fill = '#86BCB6')

ggsave("regYM.png",  bg = "white", dpi = "retina", width = 20, height = 15, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Regs")

allRegsY <- ggplot(daLong, aes(x = Year, y = Temperature)) +
  geom_point(aes(colour = City)) + facet_wrap(vars(City), nrow = 3) +
  theme_minimal() + scale_color_tableau() +
  theme(panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5),
        plot.caption = element_text(hjust = 0.5), legend.position = 'none') +
  geom_smooth(method='lm', color = '#499894', fill = '#86BCB6') +
  labs(y = 'Temperature', x = 'Year', title = "Regressions using Yearly Data",
       caption = "95% C.I. shown around fitted regression lines")

ggsave("AllRegsY.png",  bg = "white", dpi = "retina", width = 20, height = 15, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Regs")

allRegsM <- ggplot(dmLong, aes(x = Month, y = Temperature)) +
  geom_point(aes(colour = City)) + facet_wrap(vars(City), nrow = 3) +
  theme_minimal() + scale_color_tableau() +
  theme(panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5),
        plot.caption = element_text(hjust = 0.5), legend.position = 'none') +
  geom_smooth(method='lm', color = '#499894', fill = '#86BCB6') +
```

```r
  labs(y = 'Temperature', x = 'Year', title = "Regressions using Mothly Data",
       caption = "95% C.I. shown around fitted regression lines")

ggsave("AllRegsM.png",  bg = "white", dpi = "retina", width = 20, height = 15, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Regs")

allRegsYCB <- ggplot(daLong, aes(x = Year, y = Temperature)) +
  geom_point(aes(colour = City)) + facet_wrap(vars(City), nrow = 3) +
  theme_minimal() + scale_color_tableau() +
  theme(panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5),
        plot.caption = element_text(hjust = 0.5), legend.position = 'none') +
  geom_smooth(data = subset(daLong, Year < 1975),  method='lm', fullrange = F, color = '#499894', fill = '#86BCB6') +
  geom_smooth(data = subset(daLong, Year >= 1975),method='lm', fullrange = F, color = '#B07AA1', fill = '#D4A6C8') +
  labs(y = 'Temperature', x = 'Year', title = "Regressions using Yearly Data, 1975 Break",
       caption = "95% C.I. shown around fitted regression lines")

ggsave("AllRegsYCB.png",  bg = "white", dpi = "retina", width = 20, height = 15, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Regs")

allRegsYB <- ggplot(daLong, aes(x = Year, y = Temperature)) +
  geom_point(aes(colour = City)) + facet_wrap(vars(City), nrow = 3) +
  theme_minimal() + scale_color_tableau() +
  theme(panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5),
        plot.caption = element_text(hjust = 0.5), legend.position = 'none') +
  geom_smooth(data = subset(daLong, Year < 1961),  method='lm', fullrange = F, color = '#499894', fill = '#86BCB6') +
  geom_smooth(data = subset(daLong, Year >= 1961),method='lm', fullrange = F, color = '#B07AA1', fill = '#D4A6C8') +
  labs(y = 'Temperature', x = 'Year', title = "Regressions using Yearly Data, 1961 Break",
       caption = "95% C.I. shown around fitted regression lines")

ggsave("AllRegsYB.png",  bg = "white", dpi = "retina", width = 20, height = 15, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Regs")

allRegsMCB <- ggplot(dmLong, aes(x = Month, y = Temperature)) +
  geom_point(aes(colour = City)) + facet_wrap(vars(City), nrow = 3) +
  theme_minimal() + scale_color_tableau() +
  theme(panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5),
        plot.caption = element_text(hjust = 0.5), legend.position = 'none') +
  geom_smooth(data = subset(dmLong, Month < 197501),  method='lm', fullrange = F, color = '#499894', fill = '#86BCB6') +
  geom_smooth(data = subset(dmLong, Month >= 197501),method='lm', fullrange = F, color = '#B07AA1', fill = '#D4A6C8') +
  labs(y = 'Temperature', x = 'Month', title = "Regressions using Monthly Data, 1975 Break",
       caption = "95% C.I. shown around fitted regression lines")

ggsave("AllRegsMCB.png",  bg = "white", dpi = "retina", width = 20, height = 15, units = "cm",
```

```r
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Regs")

allRegsMB <- ggplot(dmLong, aes(x = Month, y = Temperature)) +
  geom_point(aes(colour = City)) + facet_wrap(vars(City), nrow = 3) +
  theme_minimal() + scale_color_tableau() +
  theme(panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5),
        plot.caption = element_text(hjust = 0.5), legend.position = 'none') +
  geom_smooth(data = subset(dmLong, Month < 196102),  method='lm', fullrange = F, color = '#499894', fill = '#86BCB6') +
  geom_smooth(data = subset(dmLong, Month >= 196102),method='lm', fullrange = F, color = '#B07AA1', fill = '#D4A6C8') +
  labs(y = 'Temperature', x = 'Month', title = "Regressions using Monthly Data, 1961 Break",
       caption = "95% C.I. shown around fitted regression lines")

ggsave("AllRegsMB.png",  bg = "white", dpi = "retina", width = 20, height = 15, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Regs")

#residuals
mod1 <- lm(da$maastricht ~ da$year)
mod2 <- lm(dm$maastricht ~ dm$month)

residplot1 <- ggplot(data = da, aes(x = mod1$residuals)) +
  geom_density(color = '#4E79A7') + #geom_histogram(fill="#4E79A7", color = 'black', position="dodge", bins = 50)+
    labs(x = 'Residuals', y = 'Frequency') +
  theme_minimal() + scale_color_tableau() +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5), legend.position = "none") +
  ggtitle("Density of Residuals, Maastricht ~ Year")

ggsave("ResidDensY.png",  bg = "white", dpi = "retina", width = 20, height = 15, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Regs")

residplot2 <- ggplot(data = dm, aes(x = mod2$residuals)) +
  geom_density(color = '#F28E2B') + #geom_histogram(fill="#4E79A7", color = 'black', position="dodge", bins = 50)+
  labs(x = 'Residuals', y = 'Frequency') +
  theme_minimal() + scale_color_tableau() +
  theme( panel.grid.minor = element_blank(), plot.title = element_text(hjust = 0.5), legend.position = "none") +
  ggtitle("Density of Residuals, Maastricht ~ Month")

ggsave("ResidDensM.png",  bg = "white", dpi = "retina", width = 20, height = 15, units = "cm",
       path = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Figures/Regs")
```

2.7 Appendix G: Source Code (Tables)

```
#######################Tables#######################
daSS <- daC[,.("De Bilt" = De.Bilt, Eelde, Maastricht)]
stargazer(daSS, out.header = F, title = "Annual Data", table.placement = "H",
          label = "BAS", out = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical
          ↪ Statistics/Tables/Descriptive/AS" )


dmSS <- dmC[,.("De Bilt" = De.Bilt, Eelde, Maastricht)]
stargazer(dmSS, out.header = F, title = "Monthly Data", table.placement = "H",
          label = "BMS",out = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical
          ↪ Statistics/Tables/Descriptive/MS" )

dmsSS <- dmsC[,.("De Bilt" = De.Bilt, Eelde, Maastricht)]
stargazer(dmsSS, out.header = F, title = "Smoothed Monthly Data", table.placement = "H",
          label = "BMSS",out = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical
          ↪ Statistics/Tables/Descriptive/MSS" )

ddSS <- ddC[,.("De Bilt" = De.Bilt, Eelde, Maastricht)]
stargazer(ddSS, out.header = F, title = "Daily Data", table.placement = "H",
          label = "BDS",out = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical
          ↪ Statistics/Tables/Descriptive/DS" )


#time series break tests
print(xtable(structtabY, align = "llll", caption = "Structural Break in Yearly Data", digits = 5,
↪ label = "BSBY"), caption.placement = 'top', table.placement = "H",
      type = "latex", file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical
      ↪ Statistics/Tables/Tests/SBY")

print(xtable(structtabM, align = "llll", caption = "Structural Break in Monthly Data", digits = 5,
↪ label = "BSBM"), caption.placement = 'top', table.placement = "H",
      type = "latex", file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical
      ↪ Statistics/Tables/Tests/SBM")

print(xtable(structtabBP, align = "llcc", caption = "Structural Break Breakpoints", digits =
↪ c(0,0,0,2), label = "BSBBP"), caption.placement = 'top', table.placement = "H",
      type = "latex", file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical
      ↪ Statistics/Tables/Tests/SBBP")

#paired t tests
print(xtable(testmat1, align = "lcccc", caption = "Paired t-tests, Yearly Data, 1961 Break",
↪ digits = c(5,3,7,4,4), label = "BPaired-t-Y"), caption.placement = 'top', table.placement =
↪ "H",
      type = "latex", file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical
      ↪ Statistics/Tables/Tests/Paired-t-Y")

```

```r
print(xtable(testmat2, align = "lcccc", caption = "Paired t-tests, Monthly Data, 1961 Break", digits = c(5,3,7,4,4), label = "BPaired-t-M"), caption.placement = 'top', table.placement = "H",
      type = "latex", file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Tests/Paired-t-M")

print(xtable(testmat21, align = "lcccc", caption = "Paired t-tests, Yearly Data, 1975 Break", digits = c(5,3,7,4,4), label = "BPaired-t-CBY"), caption.placement = 'top', table.placement = "H",
      type = "latex", file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Tests/Paired-t-CBY")

print(xtable(testmat23, align = "lcccc", caption = "Paired t-tests, Monthly Data, 1975 Break", digits = c(5,3,7,4,4), label = "BPaired-t-CBM"), caption.placement = 'top', table.placement = "H",
      type = "latex", file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Tests/Paired-t-CBM")

#t test loops
print(xtable(testmat3, align = "lccc|c|c|c", caption = "t-tests, 10-Year Means", digits = c(1,0,0,4,5,4,4), label = "BtT10Ymean"), caption.placement = 'top', table.placement = "H",
      type = "latex", file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Tests/tT10Ymean")

print(xtable(testmat4, align = "lccc|c|c|c", caption = "t-tests, 5-Year Means", digits = c(1,0,0,4,5,4,4), label = "BtT5Ymean"), caption.placement = 'top', table.placement = "H",
      type = "latex", file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Tests/tT5Ymean")

print(xtable(testmat5, align = "lccc|c|c|c", caption = "t-tests, 10-Year Medians", digits = c(1,0,0,4,5,4,4), label = "BtT10Ymedian"), caption.placement = 'top', table.placement = "H",
      type = "latex", file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Tests/tT10Ymedian")

#F-tests

print(xtable(testmat6, align = "lccccc", caption = "F-tests, Yearly Data, 1961 Break", digits = c(5,5,5,7,4,4), label = "BF-test-Y"), caption.placement = 'top', table.placement = "H",
      type = "latex", file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Tests/F-test-Y")

print(xtable(testmat7, align = "lccccc", caption = "F-tests, Monthly Data, 1961 Break", digits = c(5,5,5,7,4,4), label = "BF-test-M"), caption.placement = 'top', table.placement = "H",
      type = "latex", file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Tests/F-test-M")

print(xtable(testmat22, align = "lccccc", caption = "F-tests, Yearly Data, 1975 Break", digits = c(5,5,7,4,4,4), label = "BF-test-CBY"), caption.placement = 'top', table.placement = "H",
```

```
      type = "latex", file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Tests/F-test-CBY")

print(xtable(testmat24, align = "lccccc", caption = "F-tests, Monthly Data, 1975 Break", digits = c(5,5,5,7,4,4), label = "BF-test-CBM"), caption.placement = 'top', table.placement = "H",
      type = "latex", file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Tests/F-test-CBM")

print(xtable(testmat6, align = "lccccc", caption = "F-tests, Yearly Data, 1961 Break", digits = c(5,3,7,4,4,4), label = "BF-test-YB"), caption.placement = 'top', table.placement = "H",
      type = "latex", file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Tests/F-test-YB")

print(xtable(testmat7, align = "lccccc", caption = "F-tests, Monthly Data, 1961 Break", digits = c(5,3,7,4,4,4), label = "BF-test-MB"), caption.placement = 'top', table.placement = "H",
      type = "latex", file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Tests/F-test-MB")

print(xtable(testmat22, align = "lccccc", caption = "F-tests, Yearly Data, 1975 Break", digits = c(5,3,7,4,4,4), label = "BF-test-CBY"), caption.placement = 'top', table.placement = "H",
      type = "latex", file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Tests/F-test-CBY")

print(xtable(testmat24, align = "lccccc", caption = "F-tests, Monthly Data, 1975 Break", digits = c(5,3,7,4,4,4), label = "BF-test-CBM"), caption.placement = 'top', table.placement = "H",
      type = "latex", file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Tests/F-test-CBM")

print(xtable(testmatMan, align = "lcccc", caption = "t-tests, 1975 Break", digits = c(4,4,7,4,4), label = "Bt-t-man"), caption.placement = 'top', table.placement = "H",
      type = "latex", file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Tests/t-t-man")

#print white test results
print(xtable(testmatHsced, align = "lcc", caption = "White Tests for Heteroskedasticity", digits = c(5,5,5), label = "Bwhite"), caption.placement = 'top',
      table.placement = "H", hline.after = c(-1,0,nrow(testmatHsced),3), type = "latex", file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Regressions/white")

print(xtable(testmatHsced2, align = "lcc", caption = "White Tests for Heteroskedasticity", digits = c(5,5,5), label = "BwhiteBreak"), caption.placement = 'top',
      table.placement = "H", hline.after = c(-1,0,nrow(testmatHsced),3), type = "latex", file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Regressions/whiteBreak")


#print manual reg results
```

```
print(xtable(regMat, align = "lccc", caption = "Manually Computed Regression Coefficients, Maastricht, Yearly Data", digits = c(4,6,6,8), label = "BregMat"), caption.placement = 'top', table.placement = "H",
        type = "latex", file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Regressions/regMat")

print(xtable(regMat2, align = "lccc", caption = "Manually Computed Regression Coefficients, De Bilt, Yearly Data", digits = c(4,6,6,8), label = "BregMat2"), caption.placement = 'top', table.placement = "H",
      type = "latex", file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Regressions/regMat2")

print(xtable(regMat3, align = "lccc", caption = "Manually Computed Regression Coefficients, Eelde, Yearly Data", digits = c(4,6,6,8), label = "BregMat3"), caption.placement = 'top', table.placement = "H",
      type = "latex", file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Regressions/regMat3")

print(xtable(regMatM, align = "lccc", caption = "Manually Computed Regression Coefficients, Maastricht, Monthly Data", digits = c(4,6,6,8), label = "BregMatM"), caption.placement = 'top', table.placement = "H",
      type = "latex", file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Regressions/regMatM")

print(xtable(regMatM2, align = "lccc", caption = "Manually Computed Regression Coefficients, De Bilt, Monthly Data", digits = c(4,6,6,8), label = "BregMatM2"), caption.placement = 'top', table.placement = "H",
      type = "latex", file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Regressions/regMatM2")

print(xtable(regMatM3, align = "lccc", caption = "Manually Computed Regression Coefficients, Eelde, Monthly Data", digits = c(4,6,6,8), label = "BregMatM3"), caption.placement = 'top', table.placement = "H",
      type = "latex", file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Regressions/regMatM3")

#print full regression table

##yearly data
#full
stargazer(regYD, regYE, regYM, out.header = F, title = "Regressions, Yearly Data", table.placement = "H",
          label = "BRegY", out = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Regressions/RegY")
#restricted
stargazer(regPreBYD, regPreBYE, regPreBYM, out.header = F, title = "Regressions, Yearly Data, Before 1961 Break", table.placement = "H",
          label = "BRegYRBPre",  out = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Regressions/RegYRBPre")
```

```r

stargazer(regPostBYD, regPostBYE, regPostBYM, out.header = F, title = "Regressions, Yearly Data, After 1961 Break", table.placement = "H",
          label = "BRegYRBPost",  out = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Regressions/RegYRBPost")

stargazer(regPreCBYD, regPreCBYE, regPreCBYM, out.header = F, title = "Regressions, Yearly Data, Before 1975 Break", table.placement = "H",
          label = "BRegYRCBPre",  out = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Regressions/RegYRCBPre")

stargazer(regPostCBYD, regPostCBYE, regPostCBYM, out.header = F, title = "Regressions, Yearly Data, After 1975 Break", table.placement = "H",
          label = "BRegYRCBPost",  out = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Regressions/RegYRCBPost")

#demo
stargazer(regPreCBYM, regPostCBYM, out.header = F, title = "Regressions, Yearly Data, Before and After 1975 Break", table.placement = "H",
          column.labels = c("Year < 1975", "Year > 1975"), label = "BRegDemo",  out = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Regressions/RegDemo")

##monthly data
#full
stargazer(regMD, regME, regMM, out.header = F, title = "Regressions, Monthly Data", table.placement = "H", label = "BRegM",
          out = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Regressions/RegM")
#restricted
stargazer(regPreBMD, regPreBME, regPreBMM, out.header = F, title = "Regressions, Monthly Data, Before 1961 Break", table.placement = "H",
          label = "BRegMRBPre" ,no.space = TRUE, out = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Regressions/RegMRBPre")

stargazer(regPostBMD, regPostBME, regPostBMM, out.header = F, title = "Regressions, Monthly Data, After 1961 Break", table.placement = "H",
          label = "BRegMRBPost" ,no.space = TRUE, out = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Regressions/RegMRBPost")

stargazer(regPreCBMD, regPreCBME, regPreCBMM, out.header = F, title = "Regressions, Monthly Data, Before 1975 Break", table.placement = "H",
          label = "BRegMRCBPre", out = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Regressions/RegMRCBPre")

stargazer(regPostCBMD, regPostCBME, regPostCBMM, out.header = F, title = "Regressions, Monthly Data, After 1975 Break", table.placement = "H",
          label = "BRegMRCBPost", out = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/Regressions/RegMRCBPost")
```

```r
##Bootstrap
print(xtable(BSmat1, align = "lccc", caption = "Bootstrap: t-test for Regression Coefficients",
↪  digits = c(4,4,6,6),
             label = "BBSmat1"), caption.placement = 'top', table.placement = "H",
      type = "latex", sanitize.text.function = function(x) {x},
      file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/BS/BSmat1")

print(xtable(BSmat2, align = "lcccc", caption = "Bootstrap: Paired t-test", digits = c(4,4,6,6,6),
             label = "BBSmat1"), caption.placement = 'top', table.placement = "H",
      type = "latex", sanitize.text.function = function(x) {x},
      file = "/Users/ts/Dropbox/Apps/Overleaf/Project Mathematical Statistics/Tables/BS/BSmat2")
```