

SOP 90: Analyzing Climate Trends with R

Marc Los Huertos

August 12, 2017

Contents

1	Introduction	3
1.1	Rational	3
1.2	Trend Analysis versus Time Series Analysis	3
1.3	Generalized Steps	3
1.4	Goals for This Document	4
2	Frequentists and Bayesian Statistics	4
2.1	Bayesian Statistics	4
2.2	Frequentist Statistics	4
2.3	Examples of Frequentists Methods	4
2.4	Direct Comparision– Bayes and Frequentists	4
3	Understanding the Data: The First Step	5
3.1	Categorical versus Continuous Data	5
3.2	Types of Data Define Tests Available	5
3.3	Four Frequentists Approaches	5
3.3.1	Contingency Tables	5
3.4	Analysis of Variance	7
3.5	Linear Regression	7
3.6	Logistic Regression	7
3.6.1	Implementing Approaches in R	7
3.7	Data Quality Over Time	8
3.8	Data Source and Metadata	8
3.9	Evaluating the Structure of Data	8
3.10	Evaluating for Completeness	9
3.11	Evaluating the Central Tendencies	9
3.12	Evaluating Spread	11
3.13	'Null Hypotheses' as the Foundation of Frequentist Statistics	11
3.14	Type I and Type II Errors	11
3.15	Mathematical Mechanisms to Test Hypothesis	13

4	Linear Regression	13
4.1	What is Linear Regression?	13
4.2	Assumptions of Regression	14
4.2.1	Time Series: Ordered and Autocorrelated	14
4.3	Linear Models in R	15
4.4	Regression and Climate Change	15
4.4.1	Creating Monthly Averages of Daily Maximum Temperatures	19
4.4.2	Creating Monthly Means	19
4.5	Testing all the Months	23
4.6	Next Steps	25
4.6.1	Analyzing Minimum Daily Temperatures	25
4.6.2	Precipitation: Departure from Mean	29
4.7	Assumptions of the Linear Regression	34
4.7.1	Assumptions about ϵ	34
4.7.2	Model Diagnostics	35
5	The 'Null' Hypothesis versus Information Criteria	35
5.1	Model Comparison	35
5.2	AIC to make statements about strength of evidence	35
6	Relaxing Model Assumptions	35
6.1	Using Sources of Error in the Model	37
6.2	Generalized Least Square (GLS) and Autocorrelation	37
6.3	Adding Seasonality	37
7	Advance Modeling Approaches	37
7.1	Generalized Additive Models	37
8	Time Series Analysis	38
9	References	38

Abstract

Trend and Time Series Analyses are very important in environmental monitoring. For our purposes, developing methods to analyze climate data can using a range of tools, from relatively simple methods to advanced statistical modelling.

This document is designed to introduce a few tools to analyze regularly (i.e. daily or monthly) collected data. First, we will use a standard regression model, mixed-effects models, and finally more advanced time-series modelling approaches.

1 Introduction

1.1 Rational

In an age of industrialization and waste hazardous waste production, monitoring schemes have become ubiquitous to provide early warning and/or tracking environmental quality. To detect change or provide warning for public safety, we also need tools to assess if there are trends or if there are changes that might pose a hazard.

For our purposes, trend analysis or time series analysis is used to evaluate the contested nature of climate change – in particular, to determine if there weather changes at a regional level.

1.2 Trend Analysis versus Time Series Analysis

Trend analysis statistics are a set of tools used to detect patterns of change exceed the relative variation of the system. Detecting these changes relies on the use of mathematical models meant to describe the patterns and processes of the system in question.

Trend analysis often refers to techniques for extracting an underlying pattern of behavior in a time series which would otherwise be partly or nearly completely hidden by noise. To determine if a trend is present, e.g. if the time series is non-stationary, models are used to partition or separate the trend from other sources of variation. These sources of variation may be internal or external to the system, may have regular periods (e.g. seasons) or part of some random process (e.g. random walk). If the trend can be assumed to be linear, trend analysis can be undertaken within a formal regression analysis. If the trends have other shapes than linear, trend testing can be done by non-parametric methods, e.g. Mann-Kendall test, which is a version of Kendall rank correlation coefficient. For testing and visualization of non-linear trends, smoothing techniques are extremely valuable.

1.3 Generalized Steps

To conduct a trend analysis, it's useful to consider a pattern of steps:

1. become knowledge about the structure and limitations of the data source;
2. plot the observed data over time;
3. consider methods used to transform the data;
4. model and estimate average trend;
5. evaluate the validity of the model; and
6. interpret the trend data

1.4 Goals for This Document

This document provides EA students with developing statistical training to use various statistics to analyze climate data. To accomplish this, we explore the theoretical background and demonstrate the steps to analyze climate data. We also explain how the “null” hypothesis is used in frequentist statistics, the meaning of Type I and Type II errors, and how we draw conclusions based on the results of statistical tests. Finally, we demonstrate the steps to use selected statistical tools to analyze temperature and precipitation data collected from Bangkok, Thailand.

NOTE: I am not a statistician. But I have written this because I have never found an adequate guide for undergraduates that is both accessible and complete enough for our project. However, no guide is perfect.

First, this document is incomplete. There are hundreds of ways to analyze quantitative data and long theoretical history that supports these methods. I have selected just a few and barely delve in the theory. Second, there are areas of confusion for me and these probably are visible in the document. Even when I understand the concepts and tools, my explanation may lead to confusion. In any case, I suggest you use this as ONE resource among many (e.g. textbooks, online guides, and other faculty). In addition, if you are confused by sections, please let me know because I work hard to improve it with each iteration.

2 Frequentists and Bayesian Statistics

TBD

2.1 Bayesian Statistics

TBD

<https://www.r-bloggers.com/a-simple-intro-to-bayesian-change-point-analysis/>
<http://www.flutterbys.com.au/stats/tut/tut7.2b.html>
<https://www.r-bloggers.com/bayesian-linear-regression-analysis-without-tears-r/>

2.2 Frequentist Statistics

TBD

2.3 Examples of Frequentists Methods

TBD

2.4 Direct Comparison— Bayes and Frequentists

TBD

3 Understanding the Data: The First Step

3.1 Categorical versus Continuous Data

When we measure environmental data, they can be either continuous or categorical. Categorical data are sometimes called discrete data, e.g. count data might be considered discrete and categorical — if they are relatively number of possible values (perhaps, less than 3-4). Predictor variables can also be either categorical or continuous — where we think of values that can be integers with a great range and values between integers (i.e. values with decimals).

3.2 Types of Data Define Tests Available

Based on our understanding of the data, the choices of analysis become limited but also easier to choose from (Table

3.3 Four Frequentists Approaches

Based on these four data combination types, we will give examples of each of the four types:

3.3.1 Contingency Tables

TBD

As an example, how many consecutive days of heat waves before 1960 and afterwards in some made up city!

```
heatwave_table = matrix(c(26, 31, 45, 68, 80, 100), ncol=2)

# label the columns and rows
colnames(heatwave_table) = c('pre1960', 'post1960')
rownames(heatwave_table) = c('1', '2', '3')

fisher.test(heatwave_table)

##
## Fisher's Exact Test for Count Data
##
## data: heatwave_table
## p-value = 0.8207
## alternative hypothesis: two.sided

chisq.test(heatwave_table)

##
## Pearson's Chi-squared test
##
```

Figure 1: Decision tree for frequent inference depending on the predictor and response data types.

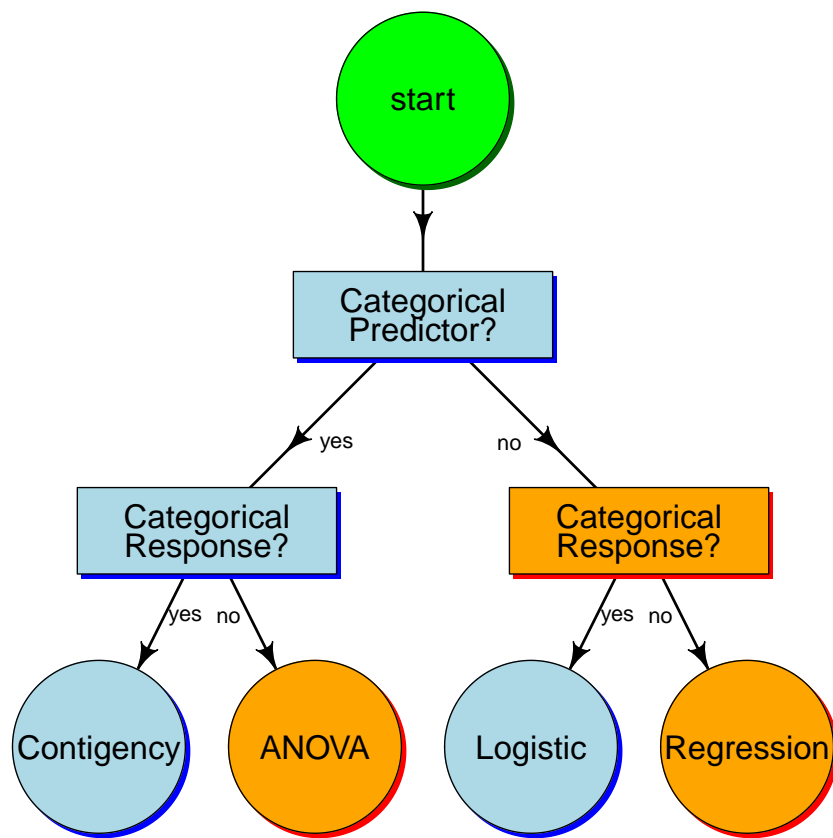


Table 1: 2 x 2 Matrix of Inference Methods –Going to insert graphics to give little pics of analysis...

		Response	
		Categorical	Continuous
Predictor	Categorical	Tests of Association fisher.test()	ANOVA aov(y ~ x), lm(y ~ x), mle(y ~ x), nlme(y ~ x)
	Continuous	Logistic Regression lm(y ~ x), glm(y ~ x), gls(y ~ x)	Linear Regression glm(y ~ x, family=binomial(link='logit'))

```
## data:  heatwave_table
## X-squared = 0.43075, df = 2, p-value = 0.8062

# convert to a table
heatwave_table = as.table(heatwave_table)

# add margins to calculate expected values!
addmargins(heatwave_table)

##      pre1960 post1960 Sum
## 1         26        68  94
## 2         31        80 111
## 3         45       100 145
## Sum      102       248 350
```

3.4 Analysis of Variance

For example, we might partition the level of warming due to anthropogenic versus natural radiative forcings:

3.5 Linear Regression

TBD

3.6 Logistic Regression

TBD

3.6.1 Implementing Approaches in R

Table 1 describes the statistical approaches available to frequentists in relation to the characteristics of the data.

In our example, the temperature is obviously continuous. Date can be treated as continuous when there are lots of them, but as describe below, it might also be considered categorical – in part because it’s ‘ordered’ and it’s divisible by day – and not finer resolution, which is decidedly not continuous. However, because there are so many in these records, we can ignore that to continue our analysis.

3.7 Data Quality Over Time

The results of all ecological studies, including time-series designs should be interpreted with caution because subtle problems can haunt the results:¹

- Data on exposure and outcome may be collected in different ways for different populations;
- Studies usually rely on routine data sources, which may have been collected for other purposes;
- Ecological studies do not allow us to answer questions about individual risks.

3.8 Data Source and Metadata

TBD

3.9 Evaluating the Structure of Data

This is analogous to selection a column or row of numbers in Excel to find the mean and you can usually find it by just looking at your spreadsheet to find the data of interest. In R you have to think a bit about what you want. Using the `str` command is good start, but we could also just look at the top of the observations to see which variables are of interest. To this we use the function `head()`, which is short for header, which shows the variable names and the first six observations.

```
head(Thailand)

##           STATION STATION_NAME      DATE  PRCP TAVG TMAX
## 1 GHCND:TH000048456 DON MUANG TH 19430101 -9999 27.6 33.9
## 2 GHCND:TH000048456 DON MUANG TH 19430102 -9999 26.8 31.7
## 3 GHCND:TH000048456 DON MUANG TH 19430103 -9999 27.2 32.8
## 4 GHCND:TH000048456 DON MUANG TH 19430104 -9999 27.3 33.3
## 5 GHCND:TH000048456 DON MUANG TH 19430105 -9999 27.8 32.2
## 6 GHCND:TH000048456 DON MUANG TH 19430106 -9999 27.1 32.8
##      TMIN      NewDate
```

¹These need to be revised...not completely appropriate.


```
## 1    NA 1943-01-01
## 2 21.7 1943-01-02
## 3 21.1 1943-01-03
## 4 21.1 1943-01-04
## 5 21.1 1943-01-05
## 6 21.1 1943-01-06
```

3.10 Evaluating for Completeness

NA is the R symbol for missing data and R requires the user to be fairly intentional about how to deal with missing data. Missing data usually mean the dataset is biased. In contrast to many software packages, R forces you to acknowledge the implications of missing data, which can be annoying, like a parent reminding you to clean your room or brush your teeth or take a shower once in the while. But the trade is worth it: you have dealt explicitly with missing data.

3.11 Evaluating the Central Tendencies

One of the first things you should do with your data is determine some of the central tendencies. For example, the mean, median, and standard deviation. Also some graphing of the data is also important. For example, what does the distribution of the data look like?

Let's start with the easy stuff. We want to get the mean of the maximum temperatures. That means we need to get the values, named TMAX from the data frame.

Okay, so we want "average." But typing average by itself doesn't show us anything except an error. Let's try `str` again. Notice the dollar symbols. These symbols are used to signify a list of values inside the data frame. To access this list, we type

```
Thailand$TMAX
```

So, now we can get the number of observations, i.e. the length of the vector, by typing

```
length(Thailand$TMAX)
## [1] 27026
```

Okay, let's calculate the mean. In this case, it requires caution. Notice there are NAs in the data.

Typing `mean(Thailand$TMAX)` gives an ambiguous result, NA. Try it. R is basically saying that the mean can not be calculated because of missing values, thus the mean is also missing. So, can we not calculate the mean when data

are missing? No, we just have to tell R what to do with missing data. In this case, we tell R to remove them, with the argument `na.rm="TRUE"`, where True can be abbreviated to T. `na.rm="TRUE"` roughly translates to 'please remove all the NAs.'

Okay as of August 12, 2017, the average is 32.9461248². It will change next month when May 2010 is added to the data set. Now let's determine the median and standard deviation.

```
median(Thailand$TMAX, na.rm=T)

## [1] 33

sd(Thailand$TMAX, na.rm=T)

## [1] 2.336892
```

If you would like a summary of each of the variables, the function is pretty easy to remember—but the output is not exceptionally pleasing.

```
summary(Thailand)
```

##	STATION	STATION_NAME	DATE
##	GHCND:TH000048456:27026	DON MUANG TH:27026	Min. :19430101
##			1st Qu.:19610848
##			Median :19800265
##			Mean :19797426
##			3rd Qu.:19980830
##			Max. :20170630
##			
##	PRCP	TAVG	TMAX
##	Min. : -9999.0	Min. : -9999.0	Min. :19.30
##	1st Qu.: 0.0	1st Qu.: 27.1	1st Qu.:31.60
##	Median : 0.0	Median : 28.4	Median :33.00
##	Mean : -1532.3	Mean : -255.2	Mean :32.95
##	3rd Qu.: 1.0	3rd Qu.: 29.6	3rd Qu.:34.40
##	Max. : 484.1	Max. : 34.4	Max. :40.80
##			NA's :5066
##	TMIN		
##	Min. : 2.40		
##	1st Qu.:23.00		
##	Median :24.50		
##	Mean :24.02		
##	3rd Qu.:25.60		
##	Max. :30.10		
##			NA's :7760
##	NewDate		
##	Min. :1943-01-01		
##	1st Qu.:1961-08-31		
##	Median :1980-02-29		
##	Mean :1980-03-04		
##	3rd Qu.:1998-08-29		
##	Max. :2017-06-30		
##			

²How many significant figures should you report? Have I reported this correctly?

Nevertheless, the output gives you a really good idea regarding the central tendencies of the entire data set. Granted typing code might seem like a major step backwards in the computer world, but after a few weeks you will appreciate not having the search through arcane menus to find which button to push—even worse, in these push-button software systems, it is often hard to figure out what they are doing. In the case of R, you have a really good idea of what it did, but were much more engaged in the process.

3.12 Evaluating Spread

When the mean and median diverge, it means that the distribution is skewed in some way. Let's see what the distribution looks like by creating a histogram.

```
hist(Thailand$TMAX)
```

The one you have made probably does not look that pretty, but with some more advanced coding, this is what it might look like (Figure 2).

Congratulations, you have made it through the next step in R! You now know how to do an exploratory analysis and even generate a basic histogram to view the distribution of a data set. Next, we use a standard statistical technique to determine the slope of the line and whether the line is statistically significant—but first we need to understand something about hypothesis testing.

3.13 'Null Hypotheses' as the Foundation of Frequentist Statistics

The null hypothesis, H_0 is the commonly accepted fact; it is the opposite of the alternate hypothesis. Researchers work to reject, nullify or disprove the null hypothesis. Researchers come up with an alternate hypothesis (H_A), one that they think explains a phenomenon, and then work to reject the null hypothesis.

The word null in this context means that it's a commonly accepted fact that researchers work to nullify. It doesn't mean that the statement is null itself! (Perhaps the term should be called the nullifiable hypothesis as that might cause less confusion).

3.14 Type I and Type II Errors

Adding to the confusion, is that there are times that the statistical test might be wrong! Researchers try to protect themselves, but as we'll learn, it's impossible to protect ourselves completely.

A type I error occurs when the null hypothesis (H_0) is true, but is rejected. It is asserting something that is absent, a false hit. A type I error may be likened to a so-called false positive (a result that indicates that a given condition is present when it actually is not present).

The type I error rate or significance level is the probability of rejecting the null hypothesis given that it is true. It is denoted by the Greek letter α (alpha)

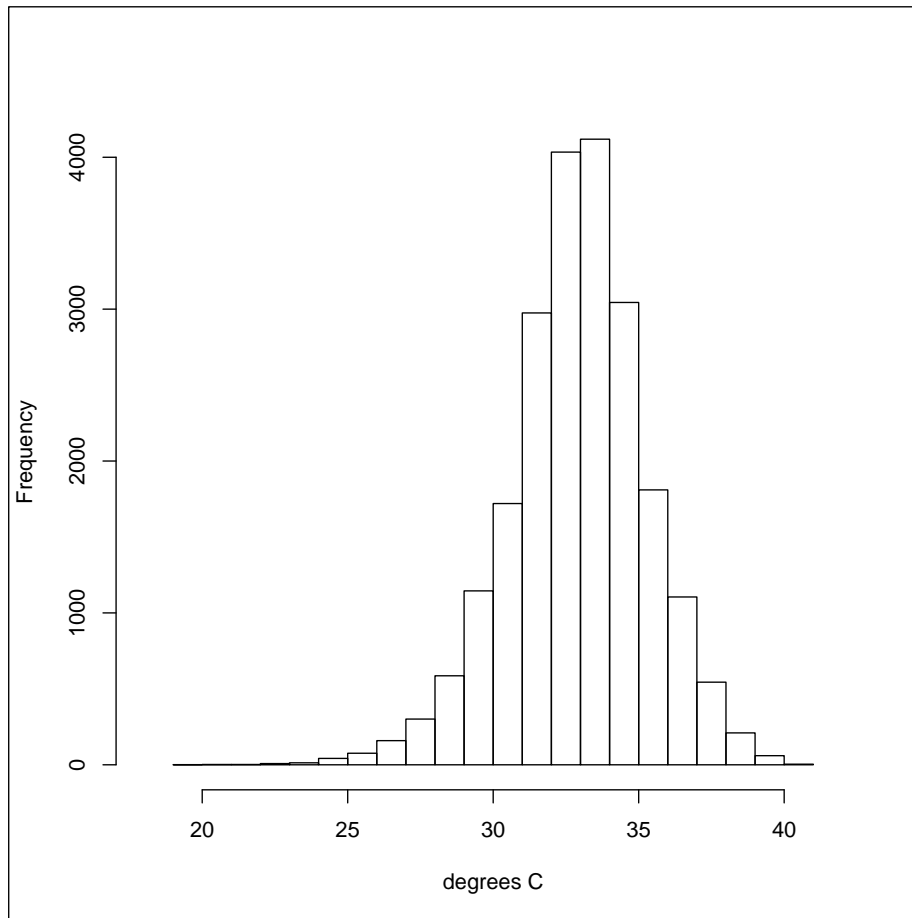


Figure 2: Histogram of Maximum Temperatures, Bangkok, Thailand.

and is also called the alpha level. Often, the significance level is set to 0.05 (5%), implying that it is acceptable to have a 5% probability of incorrectly rejecting the null hypothesis.


A type II error occurs when the null hypothesis is false, but erroneously fails to be rejected. It is failing to assert what is present, a miss. A type II error may be compared with a so-called false negative (where an actual 'hit' was disregarded by the test and seen as a 'miss') in a test checking for a single condition with a definitive result of true or false. A Type II error is committed when we fail to believe a truth. In terms of folk tales, an investigator may fail to see the wolf ("failing to raise an alarm"). Again, H_0 : no wolf.

The rate of the type II error is denoted by the Greek letter β (beta) and related to the power of a test (which equals $1-\beta$).

		Reality	
		True	False
Measured/ Perceived	True	Correct 😊	Type I False Positive
	False	Type II False Negative	Correct 😊

We can represent these choices in a table, which makes it easy to see the various outcomes and types of errors (Table 2).

Table 2: 2 x 2 Matrix of Inference Methods –Going to insert graphics to give little pics of analysis...

		Reality	
		Truth	Not True
Statistical Result	Belief	Correct Decision 	Type I
	Non-belief	Type II	Correct Decision

3.15 Mathematical Mechanisms to Test Hypothesis

Using the linear model, we can analyze several types of data, when the response variable is continuous. If the have a predictor variable that is categorical, then we often analyze the data using the method known as analysis of variance or ANOVA. If the predictor variable is continuous, then we often analyze data using a regression analysis.

4 Linear Regression

4.1 What is Linear Regression?

Linear regression is the most basic and commonly used predictive analysis. Regression estimates are used to describe data and to explain the relationship between one dependent variable and one or more independent variables. At the center of the regression analysis is the task of fitting a single line through a scatter plot. The simplest form with one dependent and one independent variable is defined by the formula:

$$y = a + b * x. \tag{1}$$

Sometimes the dependent variable is also called the response. The independent variables are also predictor variables. However, Linear Regression Analysis consists of more than just fitting a linear line through a cloud of data points. It consists of 3 stages:

1. analyzing the correlation and directionality of the data,
2. estimating the model, i.e., fitting the line, and
3. evaluating the validity and usefulness of the model.

There are three major uses for Regression Analysis: 1) causal analysis, 2) forecasting an effect, 3) trend forecasting. Other than correlation analysis, which focuses on the strength of the relationship between two or more variables, regression analysis assumes a dependence or causal relationship between one or more independent and one dependent variable.

Firstly, it might be used to identify the strength of the effect that the independent variable(s) have on a dependent variable. Typical questions are what is the strength of relationship between dose and effect, sales and marketing spending, age and income.

Secondly, it can be used to forecast effects or impacts of changes. That is, regression analysis helps us to understand how much the dependent variable will change when we change one or more independent variables. Typical questions are, How much additional Y do I get for one additional unit of X?.

Thirdly, regression analysis predicts trends and future values. The regression analysis can be used to get point estimates. Typical questions are, “What will the price for gold be 6 month from now?” “What is the total effort for a tasks?”

4.2 Assumptions of Regression

Here is a list of assumptions to produce a valid regression model:

Linear relationship TBD

Multivariate normality TBD

No or little multicollinearity TBD

No auto-correlation TBD

Homoscedasticity or Homogeneity of Variance

4.2.1 Time Series: Ordered and Autocorrelated

Time series data have a natural temporal ordering. This makes time series analysis distinct from other common data analysis problems.

4.3 Linear Models in R

The use of the linear model is the cornerstone of statistics. So ubiquitous it is rarely explained coherently. The linear model can be summarized at the equation for a line, but with the addition of error. You are probably familiar with the equation for a line where,

$$y = m * x + b \quad (2)$$

This equation defines a line, where m is the slope, b is the y-intercept, and the x and y are coordinates. The linear model is based on this form and is usually written as

$$y \sim \alpha + \beta * x + \epsilon \quad (3)$$

The order is usually changed, where the intercept is first, followed by the slope and x variable and the addition of error or noise. The error is usually symbolized as ϵ . In general, in a statistical model, Greek letters are used and instead of an equals sign, we use a tilde, meaning that that left side of the equation is a function of the right side. Luckily, this is the approximate form that R expects, so if you understand this, you will have a pretty good idea of how to code a linear model in R.

The function to build a linear model is `lm()`. This function is extremely powerful and can be easily implemented, but this is a good time to see what the help menus look like in R.

```
help(lm)
```

I am not showing it here, but you should see a long complex looking help page window pop up. All help files in R are structured the same way, so in spite of the uninterpretable text, written by and for computer programmers, the structure will become familiar. Beginning with the description, the help screen describes the function, how to use it, and give some examples. Admittedly, I rarely understand much of the text, but I find the examples to be very useful! In fact, I suggest you paste the example into R and see what happens, I find this one of the best ways to learn R. Use an example that I know works, then change it to make it do what I want it to do.

4.4 Regression and Climate Change

One of the ourcomes of the linear regression is to estimate the best fit line

$$y = mx + b + \epsilon, \quad (4)$$

where ϵ is an estimate of the error. In addition, two other estimates are provided, one for the slope, m , and the y-intercept, b .

But these estimates are also hypotheses, where the null hypothesis is:

slope is zero Rejecting the null hypothesis would be support the alternative hypothesis, or the estimate of the slope.

y-intercept is zero Rejecting the null hypothesis would support the alternative hypothesis, the estimate of the y-intercept.

Okay, let's see if we can do this for our Bangkok data. Let's test if there is a significant change of daily maximum temperatures (TMAX) with time. Thus, in general terms, Maximum temperature is a function of time, or $TMAX = f(Time)$.

$$TMAX \sim \alpha + \beta * time + \epsilon \quad (5)$$

Translating this in R will take some additional tricks besides just getting the code figured out. First, we need to identify the predictor variable, 'NewDate', in the data frame which we created in SOP85.

Because these data are in a time series, they are serially correlated, meaning that the June sample will be more like the July sample than the August sample. In addition, the June 2010 sample will be similar to the June 2009 sample. These correlation violate the assumption of independence, but for now, we will ignore this violation and just create a linear model in bliss.

For the response variable, we will use the daily maximum temperatures, TMAX. Remember there are some missing data, it will be interesting to note how R deals with that.

First, let's create a plot of data using `plot()`, whose format is `plot(x, y)` or `plot(y ~ x)`. We will use the later for now,

We use the `lm()` function that arrange the results in-line with a regression model. This syntax is straight forward,

```
lm(TMAX ~ NewDate, data=Thailand)

##
## Call:
## lm(formula = TMAX ~ NewDate, data = Thailand)
##
## Coefficients:
## (Intercept)      NewDate
##  3.289e+01      2.702e-05
```

From this model, we learn that the change in $TMAX$ is 0 degrees $year^{-1}$. Figure 4.4 shows a trend of increasing maximum temperatures.

Now determine test the null hypotheses and use the `summary()` function to display many of the important regression results.

```
summary(lm(TMAX ~ NewDate, data=Thailand))
```


Figure 3: Maximum daily temperatures for Bangkok, Thailand.

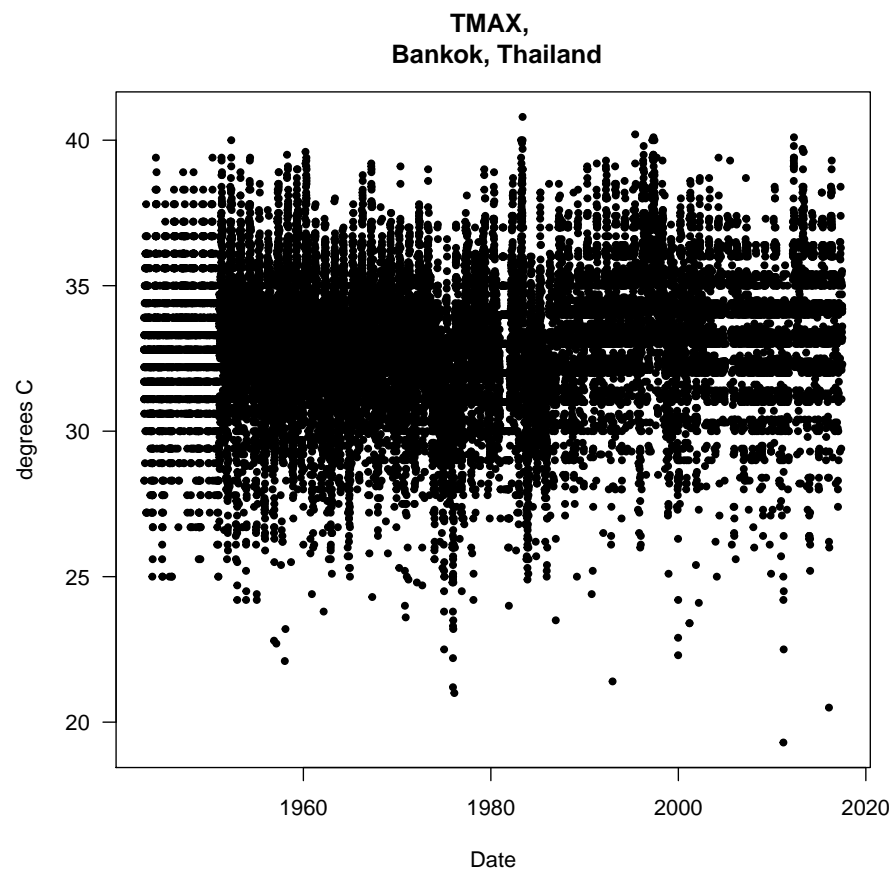
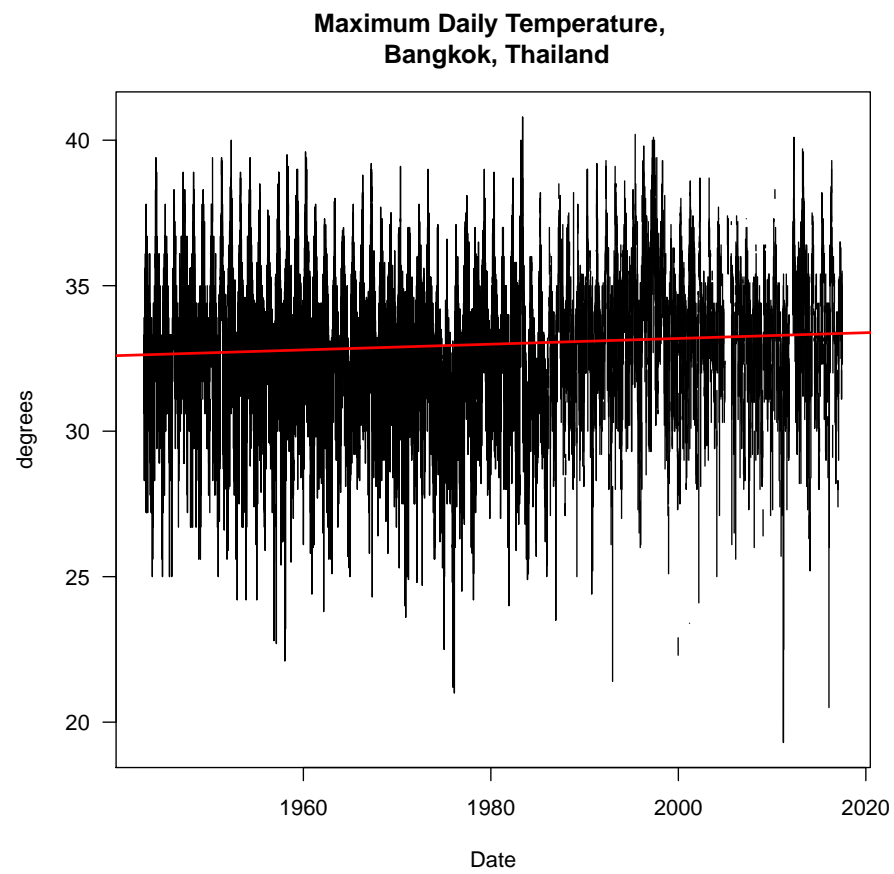


Figure 4: Maximum Daily Temperatures in Bangkok, Thailand.



```
##
## Call:
## lm(formula = TMAX ~ NewDate, data = Thailand)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.9974  -1.3193   0.0782   1.4717   7.7771
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.289e+01  1.631e-02 2016.35  <2e-16 ***
## NewDate      2.702e-05  2.140e-06  12.62  <2e-16 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.329 on 21958 degrees of freedom
## (5066 observations deleted due to missingness)
## Multiple R-squared:  0.007202, Adjusted R-squared:  0.007157
## F-statistic: 159.3 on 1 and 21958 DF,  p-value: < 2.2e-16
```

Based on the results, we reject the null hypotheses, i.e. the events that this might occur by chance is small: 2×10^{-16} for the slope is zero and $p \leq 2 \times 10^{-16}$ for the y-intercept is zero.

In addition, we have some information on the residuals, and R^2 estimates, which are important to interpret the model.

For now, we can appreciate the the temperature is changing, i.e. increasing, with a slope of 2.7×10^{-5} degrees C per year.

4.4.1 Creating Monthly Averages of Daily Maximum Temperatures

One of the first things to note is how messy the data look and there are lots of sources of variation. For example, we expect months to respond differently to the climate change. To assess this, we will now analyze the data for monthly means of the maximum temperatures.

4.4.2 Creating Monthly Means

To create monthly means, we need to disaggregate the NewDate variable into a month and year variables.

First we can use the `as.Date()` function to extract a portion of the date, where `%m` is for month and `%Y` is for a four digit year. Then, we create new variables in our dataframe, one for month and one for year.

```
Thailand$Month = format(as.Date(Thailand$NewDate), format = "%m")
Thailand$Year = format(Thailand$NewDate, format="%Y")
```

After creating the month and year as separate variables, we can use them to calculate the mean using the `aggregate()` function. In the code below, we can also calculate the standard deviation too, although I haven't used this measure in this document, several students have asked for this for their analysis.

```
MonthlyTMAXMean = aggregate(TMAX ~ Month + Year, Thailand, mean)

MonthlyTMAXMean$YEAR = as.numeric(MonthlyTMAXMean$Year)
MonthlyTMAXMean$MONTH = as.numeric(MonthlyTMAXMean$Month)
str(MonthlyTMAXMean)

## 'data.frame': 888 obs. of 5 variables:
## $ Month: chr  "01" "02" "03" "04" ...
## $ Year : chr  "1943" "1943" "1943" "1943" ...
## $ TMAX : num  32.2 33.2 34.9 33.5 33.8 ...
## $ YEAR : num  1943 1943 1943 1943 1943 ...
## $ MONTH: num  1 2 3 4 5 6 7 8 9 10 ...

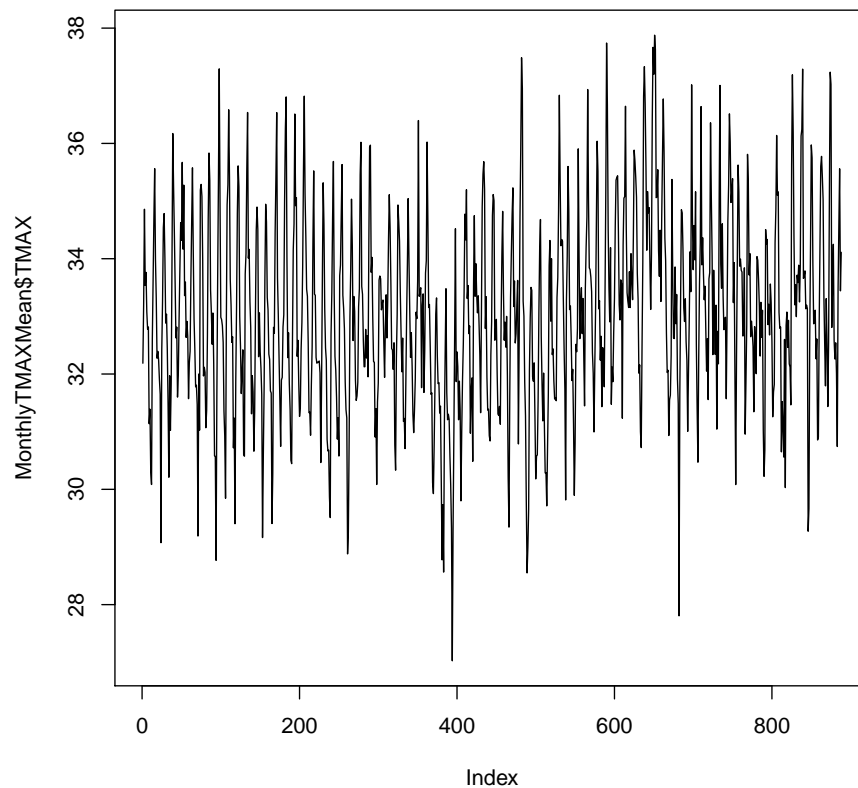
MonthlyTMAXSD = aggregate(TMAX ~ Month + Year, Thailand, sd)

MonthlyTMAXSD$YEAR = as.numeric(MonthlyTMAXSD$Year)
MonthlyTMAXSD$MONTH = as.numeric(MonthlyTMAXSD$Month)
MonthlyTMAXSD$NewDate = MonthlyTMAXSD$YEAR + (MonthlyTMAXSD$MONTH - 1)/12

head(MonthlyTMAXSD)

##   Month Year      TMAX YEAR MONTH  NewDate
## 1    01 1943 1.523317 1943      1 1943.000
## 2    02 1943 1.668648 1943      2 1943.083
## 3    03 1943 1.969395 1943      3 1943.167
## 4    04 1943 2.521970 1943      4 1943.250
## 5    05 1943 2.100818 1943      5 1943.333
## 6    06 1943 1.041763 1943      6 1943.417

plot(MonthlyTMAXMean$TMAX, ty='l')
```



Below is Standard Deviation

```
#plot(MonthlySD$TMAX, ty='l')

#plot(TMAX~ NewDate, data=MonthlySD, ty='l')
#SD.lm <- lm(TMAX~NewDate, data=MonthlySD)
#summary(SD.lm)

#abline(coef(SD.lm), col="red")
```

Selecting for 1 Month – May

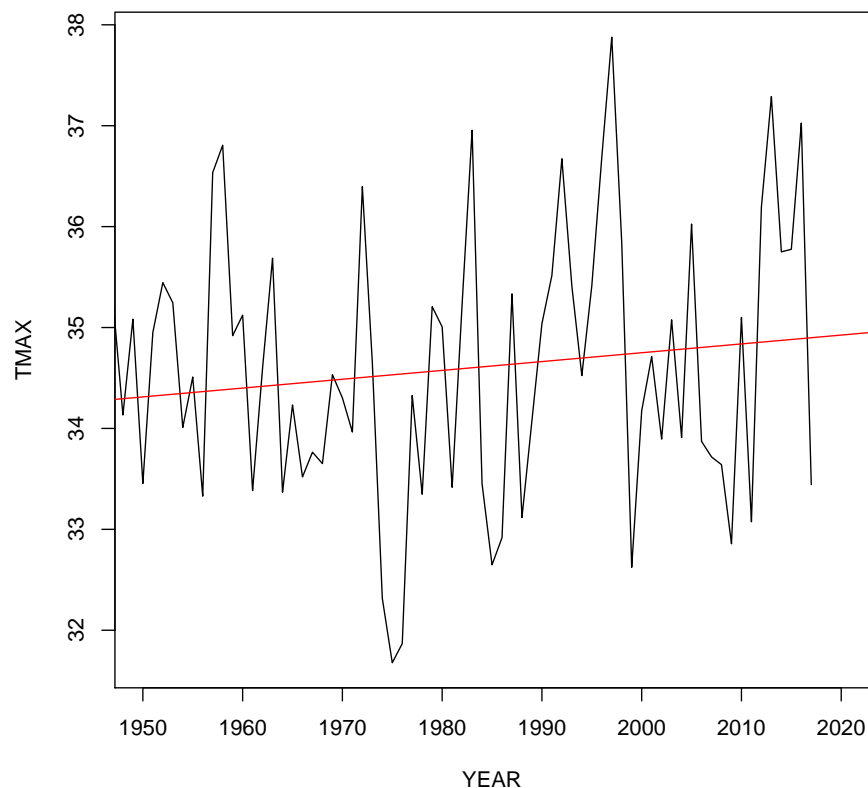
Perhaps, we can get a better handle on this stuff if we analyze for just one month at a time – certainly easier to visualize!

```
#plot(MonthlyTMAXMean$TMAX[MonthlyTMAXMean$Month=="05"], ty='l')
plot(TMAX~YEAR, data=MonthlyTMAXMean[MonthlyTMAXMean$Month=="05",], ty='l', xlim=c(1950, 2020))
May.lm <- lm(TMAX~YEAR, data=MonthlyTMAXMean[MonthlyTMAXMean$Month=="05",])
```

```
summary(May.lm)

##
## Call:
## lm(formula = TMAX ~ YEAR, data = MonthlyTMAXMean[MonthlyTMAXMean$Month ==
##      "05", ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.85376 -0.93210 -0.04633  0.81231  3.15347
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.238621  13.753642   1.253   0.214
## YEAR         0.008756   0.006946   1.261   0.211
##
## Residual standard error: 1.302 on 73 degrees of freedom
## Multiple R-squared:  0.0213, Adjusted R-squared:  0.007897
## F-statistic: 1.589 on 1 and 73 DF, p-value: 0.2115

abline(coef(May.lm), col="red")
```



Now, the change is 0.0088 degree C/year or 0.876 degree C/100 years with a probability of 0.2115. Although we can't reject the null hypothesis, we find the method to be fairly straightforward!

4.5 Testing all the Months

I think you should evaluate every month and see what happens. You might also consider looking at the TMIN as well. Could be important!³

Below, I have create code to evaluate all of the months at once, but you may prefer to go through each month manually and change the number from 5 to other months of the year.

³What about multiple hypotheses in one dataset!

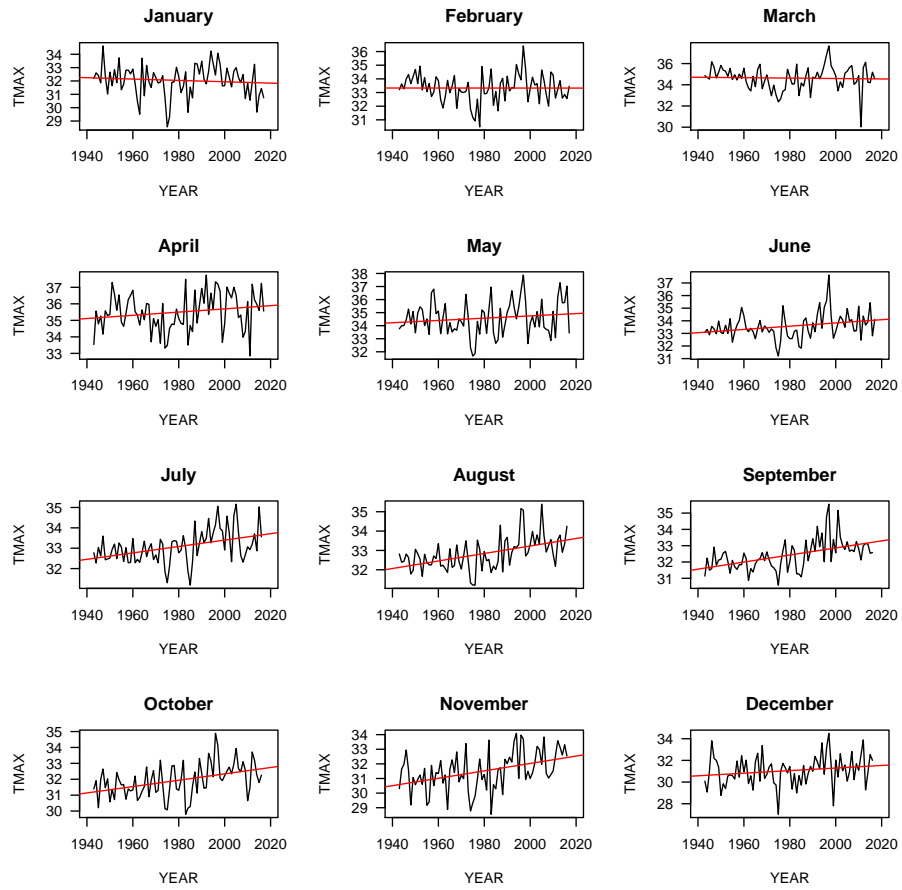
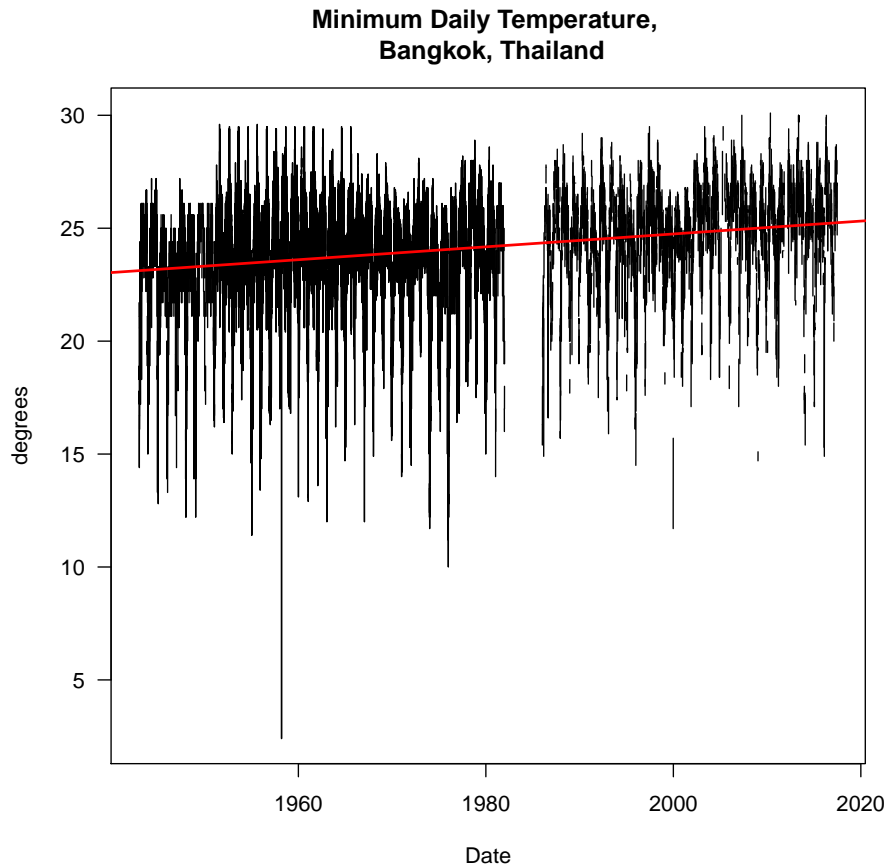


Figure 5: Minimum Daily Temperatures in Bangkok, Thailand.



4.6 Next Steps

4.6.1 Analyzing Minimum Daily Temperatures

Alternatively, it might be important to evaluate changes to the daily minimum temperatures. Following the same steps we used before but using the TMIN instead of TMAX, let's analyze the monthly average of daily minimum temperatures by following these steps:

1. First, let's plot the daily minimum temperatures, and as with the daily maximum temperatures, find tons of scatter (Table 1).
There appears to be a trend, but it's clouded with lots of variation.
2. We create a monthly TMIN mean for each month.

```

MonthlyTMINMean = aggregate(TMIN ~ Month + Year, Thailand, mean)

MonthlyTMINMean$YEAR = as.numeric(MonthlyTMINMean$Year)

# Fixing the Format of Month and Year as numeric
MonthlyTMINMean$YEAR = as.numeric(MonthlyTMINMean$Year)
MonthlyTMINMean$MONTH = as.numeric(MonthlyTMINMean$Month)
head(MonthlyTMINMean)

##      Month Year      TMIN YEAR MONTH
## 1      01 1943 18.54828 1943      1
## 2      02 1943 20.73077 1943      2
## 3      03 1943 23.39655 1943      3
## 4      04 1943 23.79259 1943      4
## 5      05 1943 24.87692 1943      5
## 6      06 1943 24.76429 1943      6

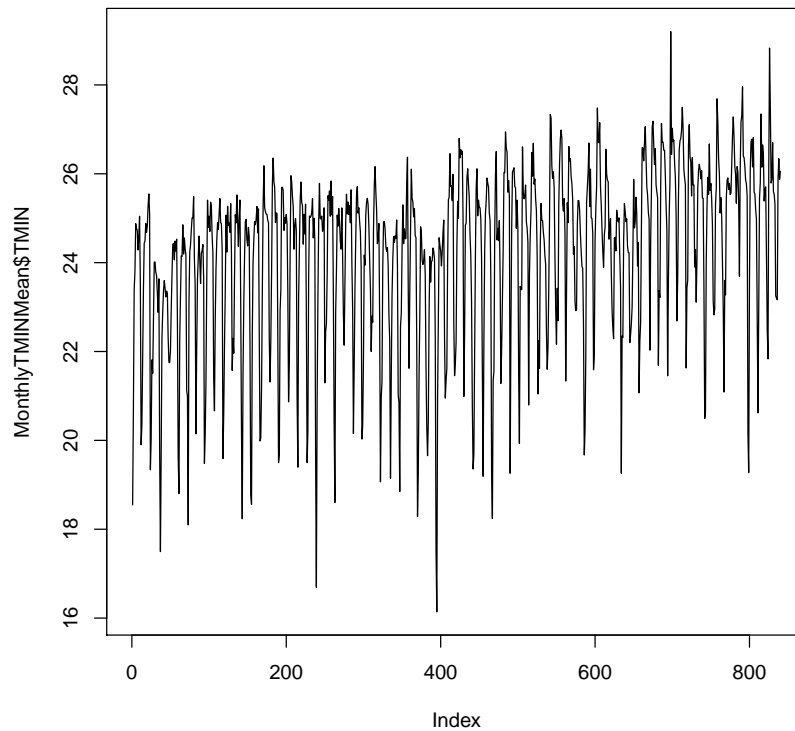
```

3. Create a plot of the monthly average of the daily minimum temperatures.

```

plot(MonthlyTMINMean$TMIN, ty='l')

```

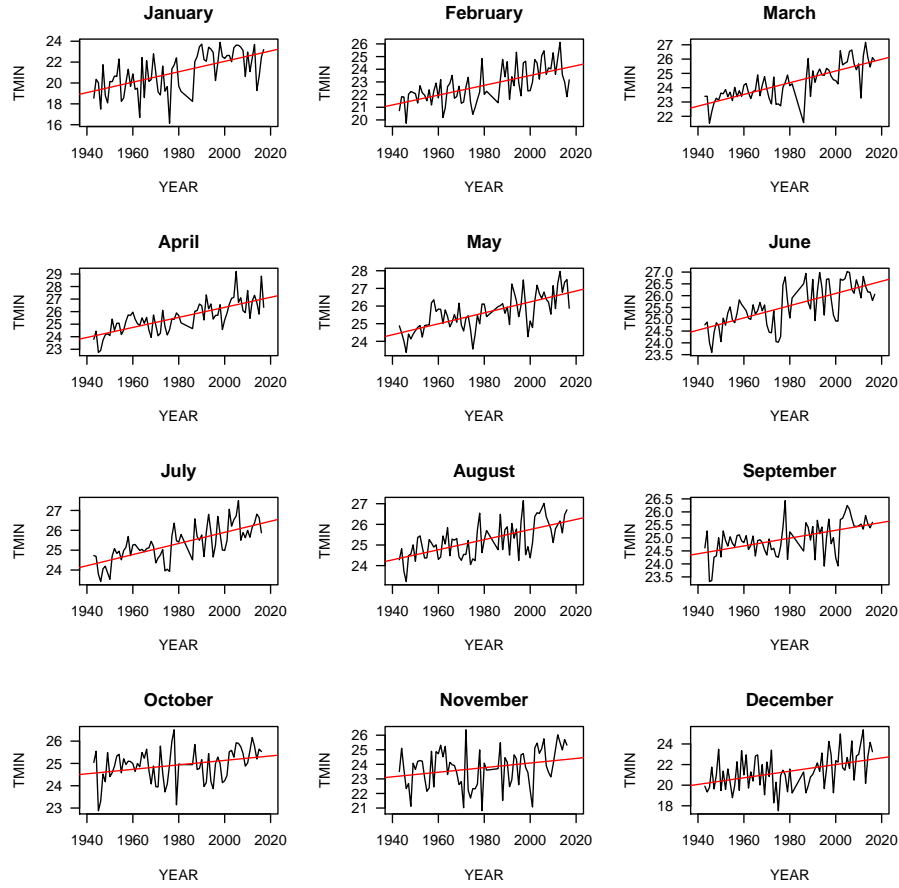


There is still lots of scatter and now we can subset our data by month.

4. Using the example above, we'll plot all 12 months at once to look for patterns (Table 6).
5. The change in minimum temperatures seems to be even more compelling than the maximum temperatures. To compare, look at the Table ?? to appreciate estimated slopes and their associated null hypothesis probabilities.

```
library(xtable)
Results <- data.frame(Month = TMINresult[c(2:13),1], TMINslope = TMINresult[c(2:13),2],
Results$starTMIN = "NS"
Results$starTMIN[Results$TMIN_P <= .05] = "*"
Results$starTMIN[Results$TMIN_P < 0.01] = "**"
Results$starTMIN[Results$TMIN_P < 0.001] = "***"
Results$starTMAX = "NS"
Results$starTMAX[Results$TMAX_P < 0.05] = "*"
Results$starTMAX[Results$TMAX_P < 0.01] = "**")
```

Figure 6: Twelve Months of Monthly Average Daily Minimum Temperatures, Bangkok, Thailand



```
Results$starTMAX[Results$TMAX_P < 0.001] = "***"
Results$TMINSlope=paste(Results$TMINSlope, Results$starTMIN)
Results$TMAXslope=paste(Results$TMAXSlope, Results$starTMAX)
colnames(Results) <- c("Month", "2", "3", "R^2", "5", "6", "R^2", "8", "9", "Slope TMIN")
print(xtable(Results[,c(1, 10, 4, 11, 7)]))
```

Based on the results above, the slopes are greatest during the dry season (starting in May) for the maximum temperatures – but the minimum temperatures show the largest slopes (change) and peaking between January and April.

In addition, the r^2 values signify the amount of the variance explained by the predictor – in the case of TMIN, most of the values are over 20%

	Month	Slope TMIN	R ²	Slope TMAX	R ² .1
1	January	0.0498 ***	0.341	-0.0051 NS	0.009
2	February	0.0387 ***	0.385	-1e-04 NS	0
3	March	0.0409 ***	0.567	-0.002 NS	0.001
4	April	0.04 ***	0.551	0.0097 NS	0.035
5	May	0.031 ***	0.48	0.0088 NS	0.021
6	June	0.0259 ***	0.448	0.0129 *	0.078
7	July	0.028 ***	0.493	0.0157 ***	0.173
8	August	0.0245 ***	0.395	0.0194 ***	0.245
9	September	0.015 ***	0.279	0.0217 ***	0.257
10	October	0.01 *	0.093	0.02 ***	0.174
11	November	0.0158 *	0.067	0.0253 ***	0.169
12	December	0.0322 ***	0.178	0.0119 NS	0.034

meaning that over 20% of the variance is explained by time. While in March and April over time explains 50% of the variance.

This is very high for uncontrolled experiments. However, we should be cognizant that in many cases, especially for the maximum temperatures, it is less than 10%. This means the the variation in temperature are not predicted by time – thus, as a modeler, I would work very hard to capture other sources to better understand what is going on in Thailand.

Finally, we should also be very concerned about testing 2 dozen hypotheses with our little R code. It's easy to do, but based on chance alone, with a critical value of 0.05, we should expect 1 in 20 tests to give us a Type I error, a signal when one doesn't exist. Since we did 12 tests, we should expect a good chance that one or more of our tests will reject the null hypothesis incorrectly. Yikes! Please keep this in mind and be careful to avoid this potential problem.

As we might expect, the a small amount of the variance is explained by the “Month.” Many things predict temperature, that year is one, is quite problematic.

6. What we have not determined is the cause. So, be careful when you describe the results, cause and effect cannot be analyzed using this method.

4.6.2 Precipitation: Departure from Mean

Precipitation might depend more on the departure from the mean (often referred as as normal, whatever that means!). I think it's worth pursuing, but haven't finished the analysis yet.

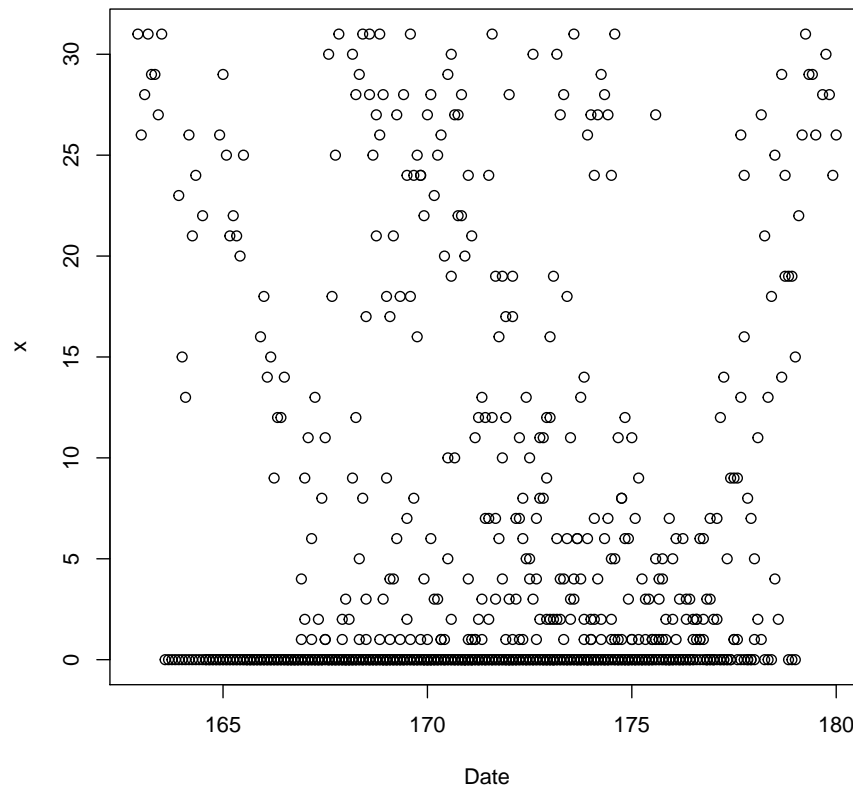
Precipitation is something that might increase or decrease due to climate change. So, to analyze this, we will evaluate how much precipitation has deviated from the mean, by plotting the rainfall and the mean in a time-series plot.

Second, we need to remove the missing values and evaluate which years have complete years. If you are missing rainy months, then the whole year should be thrown out – but what about partial years in the drought season? We'll need to be consistent – assuming that missing data are not zeros, we'll define complete years as over 300 days of data.

NOTE: The missing values have not been converted to NAs!

```
Thailand$PRCP[Thailand$PRCP==--9999] <- NA

Missing <- aggregate(is.na(Thailand$PRCP), list(Thailand$Month, Thailand$Year), sum)
Missing$Date = as.numeric(Missing$Group.1) + as.numeric(Missing$Group.2)/12
plot(x ~ Date, data=Missing)
```



Third, we will need to decide what level of aggregation – monthly, yearly, etc. Let's aggregate by month and year to get monthly totals.

There are loads of missing values in many months. Let's cut off the months that have more than 4 missing days.

```

TotalPPT <- aggregate(Thailand$PRCP, list(Thailand$Month, Thailand$Year), sum, na.rm=T)
names(TotalPPT) = c("Group.1", "Group.2", "ppt")
NonMissing <- Missing[Missing$x < 5, c(1:3)]
library(dplyr)

##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

PPT <- merge(TotalPPT, NonMissing, all.y=TRUE)
PPT$Date <- as.numeric(PPT$Group.1) + as.numeric(PPT$Group.2)/12
head(PPT)

##   Group.1 Group.2 ppt x      Date
## 1      01     1951 0.2 0 163.5833
## 2      01     1952 0.0 0 163.6667
## 3      01     1953 20.3 0 163.7500
## 4      01     1954  5.3 0 163.8333
## 5      01     1955  2.2 0 163.9167
## 6      01     1956  5.6 0 164.0000

```

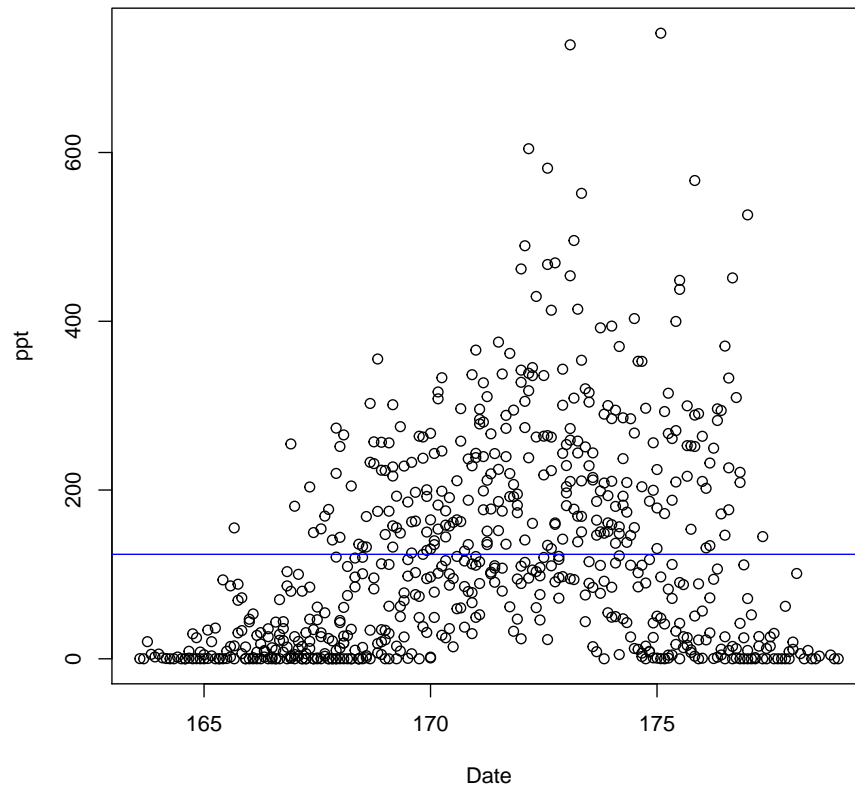
First, we need a "mean" – The IPCC uses 1961-1990 as a norm for temperature, I don't know what is the standard for rainfall or Thailand, so we should look that up. For now, we'll use our filtered records to generate a mean.

```
PRCP_mean = mean(PPT$ppt)
```

```

plot(ppt~Date, data=PPT)
abline(h=PRCP_mean, col="blue")

```

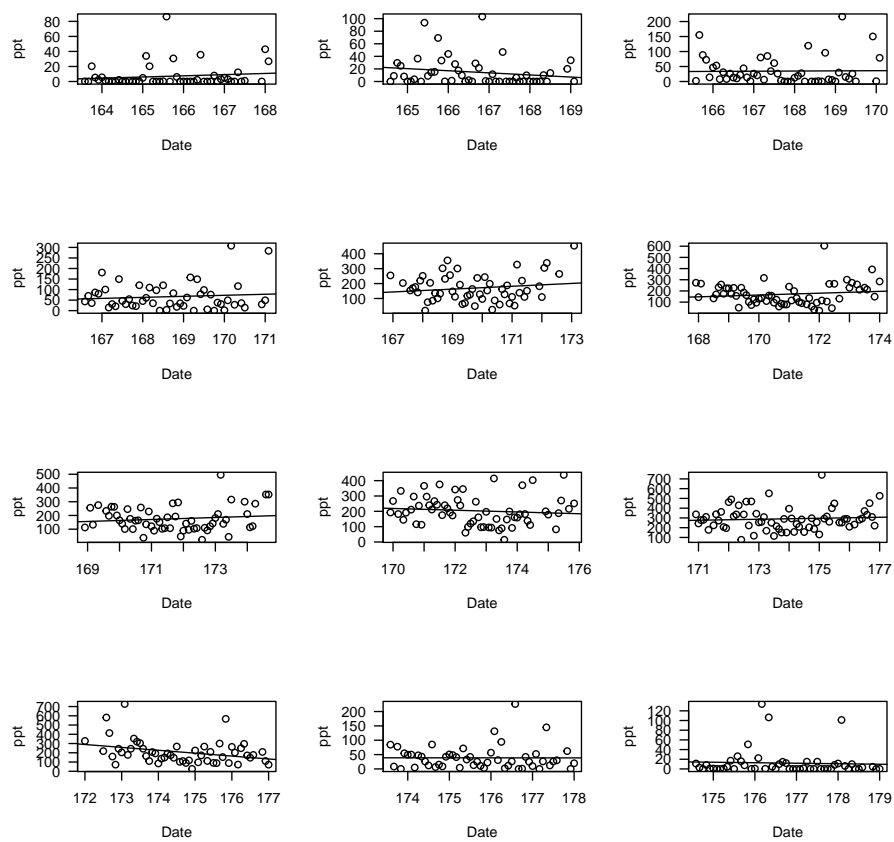


Wow, these data look terrible – the mean looks meaningless given the biased data set. I don't think we can do more analysis with this. But let's look at a few months and see what we can decipher.

```
#LosAngelesfPRCP[LosAngelesfPRCP==9999] <- NA
#YearlySum = aggregate(PRCP ~ Year, LosAngeles, sum)
#YearlySumfYEAR = as.numeric(YearlySumfYear)
#YearlyMean = mean(YearlySumfPRCP)
```

A yearly mean, based on the annual sum for the entire records. Not sure this is appropriate.

Figure has points of the yearly sum of rainfall and the blue line mean. The greenline is the trend and red line is a five year running average, I think! I am still trying to understand what the code is doing.



```
#plot(PRCP~YEAR, data=YearlySum, las=1, ty="p")
#abline(h=YearlyMean, col="blue")
#YearlySum.lm = lm(PRCP~YEAR, data=YearlySum)
#abline(coef(YearlySum.lm), col="green")

#n <- 5
#k <- rep(1/n, n)
#k

#y_lag <- stats::filter(YearlySum$PRCP, k, sides=1)
#lines(YearlySum$YEAR, y_lag, col="red")

#summary(YearlySum.lm)
```

4.7 Assumptions of the Linear Regression

Regression models, like all statistics, rely on certain assumptions. Violations of these assumptions reduces the validity of the model. If the violations are serious, then the model could be misleading or even incorrect.

TBD

4.7.1 Assumptions about ϵ

The error term should have

$E(\epsilon_t) = 0$, zero mean

$E(\epsilon_t) = s$, constant variance

$E(\epsilon_t, X_t) = 0$, no correlation with X

TBD $E(\epsilon_t, \epsilon_{t-1})$, no autocorrelation.

ϵ Normally distributed (for hypothesis testing).

Assumption four is especially important and most likely not to be met when using time series data.

Autocorrelation.

1. It is not uncommon for errors to track themselves; that is, for the error at time t to depend in part on its value at $t - m$, where m is a prior time period.

4.7.2 Model Diagnostics

With every statistical test done, researchers validate their model in some way or another. Often this entails the use of diagnostics, a standardized battery of procedures to check to see if the data are following the assumptions.

In R four plots are created by default. To see them all at the same time, we need to change the graphical parameters, using the `par()` function. In this case, we use `par(mfrow=c(2,2))` to create alter the graphics window expects four panels, in this case a 2 rows and two columns.

Try not to get bogged down in the code at this point. But noting this function can be handy in a number of ways to improve one's graphics.

To determine the validity of linear model assumptions (e.g. normality or heterogeneity of variance), you have probably used statistical tests; in contrast statisticians almost exclusively look at diagnostic plots. Why? When assumptions are violated the tests to determine violations do not perform well. So, let's see how to look at these assumptions graphically with these diagnostic plots. Linear models should have diagnostic plots that do not have any obvious structure or pattern. In this case, Figure 4.7.2 should show a great deal remaining structure in the residuals. Although for today, we are not going to try to interpret these figures, but you should notice there is a ton of unaccounted structure, i.e. variance, in the model. This is due, in part, to a violation of independence; these data are serially correlated and the model does not account for that and is inappropriate because of this. It also appears that a straight-line model does not fit well and a curvilinear should be investigated.

A properly specified model is shown in

5 The 'Null' Hypothesis versus Information Criteria

TBD

5.1 Model Comparison

TBD

5.2 AIC to make statements about strength of evidence

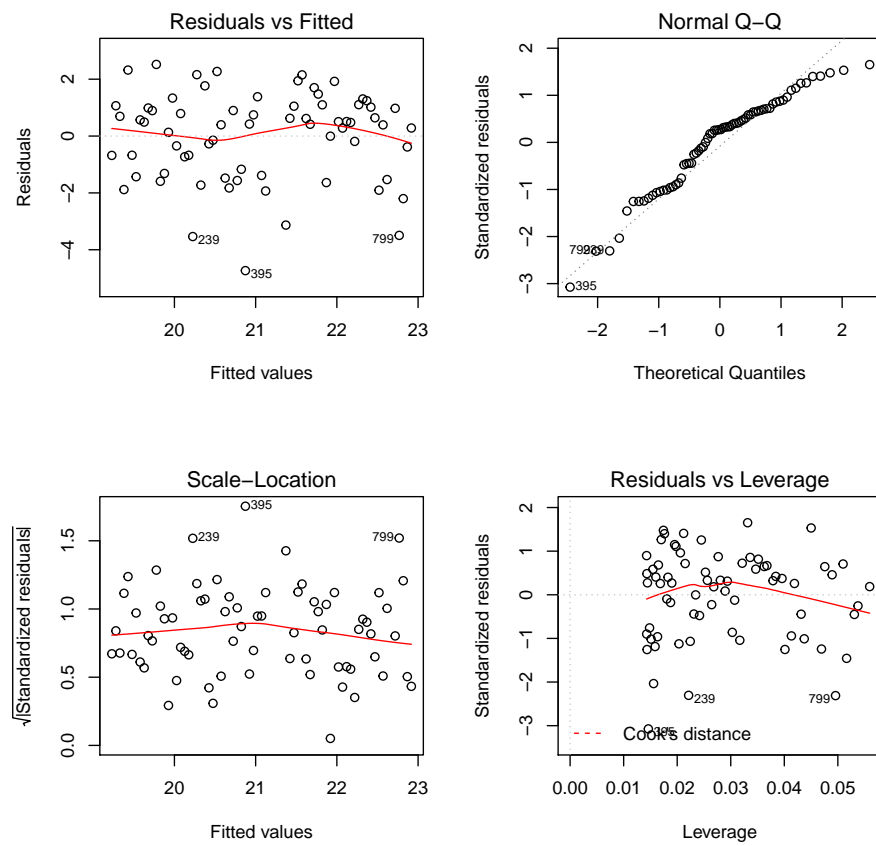
TBD

6 Relaxing Model Assumptions

TBD

Figure 7: Default diagnostic plots for a linear model in R.

```
par(mfrow=c(2,2))
plot(lm(TMIN ~ YEAR, data=MonthlyTMINMean[MonthlyTMINMean$MONTH==1,]))
```



6.1 Using Sources of Error in the Model

Instead of letting autocorrelation be 'hidden' problem in the data, we can incorporate the correlation structure into the model and use it to our advantage – create a better, i.e. unbiased estimate of the model parameters.

6.2 Generalized Least Square (GLS) and Autocorrelation

```
library(nlme)

##
## Attaching package: 'nlme'
## The following object is masked from 'package:dplyr':
##
## collapse

#TMAX.gls = gls(TMAX ~ NewDate, data = Thailand, na.action=na.omit)
#summary(TMAX.gls)
#TMAX.gls2 = gls(TMAX ~ NewDate, data = Thailand, correlation = corAR1(form=~1), na.action=na.omit)
#summary(TMAX.gls2)

#anova(TMAX.gls, TMAX.gls2)
```

6.3 Adding Seasonality

TBD

7 Advance Modeling Approaches

TBD

7.1 Generalized Additive Models

You may want to examine the GAM package in R, as it can be adapted to do some (or all) of what you are looking for. The original paper (Hastie & Tibshirani, 1986) is available via OpenAccess if you're up for reading it.

Essentially, you model a single dependent variable as being an additive combination of 'smooth' predictors. One of the typical uses is to have time series and lags thereof as your predictors, smooth these inputs, then apply GAM.

This method has been used extensively to estimate daily mortality as a function of smoothed environmental time series, especially pollutants. It's not OpenAccess, but (Dominici et al., 2000) is a superb reference, and (Statistical Methods for Environmental Epidemiology with R) is an excellent book on how to use R to do this type of analysis.

8 Time Series Analysis

TBD

9 References