

Data Analytic and Empirical Forecasting Methods using Smart Linear Regression

Don McNeil

Emeritus Professor, Macquarie University, Australia

Prince of Songkla University, Thailand, 23 January 2022

National Aeronautics & Space Administration (NASA) Data

The *Terra* Satellite

Downloading Land Surface Temperature (LST) Data

Data Structure & Study Design

Linear Regression Models and Spline Forecasts

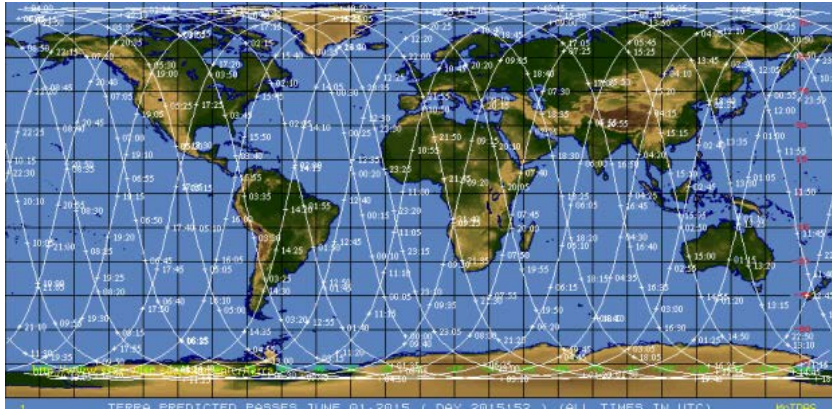
Correlation and Multivariate Regression

Graphs of Models with Forecasts

Schematic Maps of Regions and Sub-Regions

Take-home Message





We'll use remote sensing Land Surface Temperature (LST) data recorded by NASA satellites to illustrate data analytic methods. The data are free and cover all land on our planet.

Terra



Terra (EOS AM-1)

“MODIS (or Moderate Resolution Imaging Spectroradiometer) is a key instrument aboard the **Terra (EOS AM)** and **Aqua (EOS PM)** satellites. Terra's orbit around the Earth is timed so that it passes from north to south across the equator in the morning, while Aqua passes south to north over the equator in the afternoon. Terra MODIS and Aqua MODIS are viewing the entire Earth's surface every 1 to 2 days, acquiring data in 36 spectral bands, or groups of wavelengths (see MODIS Technical Specifications). These data will improve our understanding of global dynamics and processes occurring on the land, in the oceans, and in the lower atmosphere. MODIS is playing a vital role in the development of validated, global, interactive Earth system models able to predict global change accurately enough to assist policy makers in making sound decisions concerning the protection of our environment.”

(<http://modis.gsfc.nasa.gov/about/>)



NATIONAL AERONAUTICS
AND SPACE ADMINISTRATION

Mission type	Climate research
Operator	NASA
COSPAR ID	1999-068A
SATCAT №	25994
Website	terra.nasa.gov
Spacecraft properties	
Manufacturer	NASA
Launch mass	4,864 kilograms (10,723 lb)
Start of mission	
Launch date	December 18, 1999, 18:57:39 UTC
Rocket	Atlas IIAS AC-141
Launch site	Vandenberg SLC-3E

[https://en.wikipedia.org/wiki/Terra_\(satellite\)](https://en.wikipedia.org/wiki/Terra_(satellite))

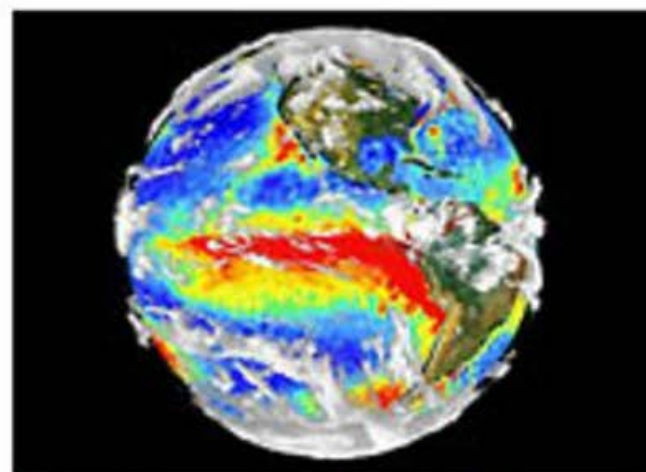
Overview of NASA's Terra satellite

John Maurer

University of Hawai'i at Mānoa
1680 East-West Rd., POST-815C
Honolulu, HI 96822, USA
Email: jmaurer@hawaii.edu

November, 2001

As an employee at the National Snow and Ice Data Center (NSIDC) from 2001-2009, I was part of an organization responsible for archiving, distributing, and supporting certain snow and ice related data products from the Moderate-Resolution Imaging Spectroradiometer (MODIS) instrument aboard NASA's Terra satellite, which was launched December 18, 1999. Providing a suite of earth observations of various sorts, one objective of MODIS is to map snow cover and sea ice across the globe on a daily basis to help scientists assess and better predict global warming. This site provides an overview of the Terra satellite and its five sensors (ASTER, CERES, MISR, MODIS, and MOPITT), which I have put together from various online sources.



Global observations from the Terra satellite.

www2.hawaii.edu/~jmaurer/terra/

Climate warming, rising sea level, deforestation, desertification, ozone depletion, acid rain, and reduction of biodiversity are all examples of ongoing global environmental change that are increasingly affecting our planet. The well-being of human beings and life at large may become largely dependent on our ability to understand the factors behind these events so that we can predict future impacts and take appropriate action to prevent them from getting any worse. Scientific research on stratospheric ozone in the 1970's, for example, led to the 1987 [Montreal Protocol](#) for worldwide reduction in production of chlorofluorocarbons (CFCs). Causes for global change may be natural as well as human-induced, furthermore, and may be persistent and long-term or just part of a normal climatic cycle: it will be important to distinguish between these different hypotheses. NASA's Earth Science Enterprise ([ESE](#)) is a Presidential Initiative supported by Congress to promote a better understanding of global environmental change using space-, ground-, and aircraft-based measurements. ESE became an official program in 1990 and is NASA's contribution to the U.S. Global Change Research Program ([USGCRP](#)), which is the United States' part in the larger worldwide effort to study global change.

The Earth Observing System (EOS) is the centerpiece of ESE, and Terra is the "flagship" of EOS. Mission planning for EOS began as far back as 1982. The program consists of a science segment, a data system, and a space segment to support a series of polar-orbiting satellites. The EOS Data and Information System (EOSDIS) is currently composed of eight Distributed Active Archive Centers (DAACs) located around the country who are responsible for processing, archiving, and distributing EOS data. All EOS data can be ordered through the EOS Data Gateway (EDG) as they become available and are, with a few exceptions, currently *free to the public*. The program is slated to continue until at least 2015 and in conjunction with the international community.

The Terra satellite was launched on December 18, 1999 and began collecting data on February 24, 2000. It operates in a polar sun-synchronous orbit at 705 km above the Earth's surface, crossing the equator on descending passes at 10:30 AM, when daily cloud cover is typically at a minimum over land. Because of this morning equatorial crossing time, "Terra" (a mythical name for "Mother Earth") was originally named EOS-AM-1. Terra has a repeat cycle of 16 days, meaning every 16 days it crosses the same spot on the Earth. It is roughly the size of a small school bus. Follow-on missions are planned to continue key measurements made by the five instruments aboard Terra: ASTER, CERES, MISR, MODIS, and MOPITT.

To download MODIS data, go to <https://modis.ornl.gov/globalsubset/>.

The screenshot shows the top section of the MODIS/VIIRS Subsets website. At the top, there's a dark blue header with the NASA logo, 'EARTHDATA', and 'Other DAACs'. Below this is a large banner with a satellite view of Earth. On the left, it says 'ORNL DAAC' and 'MODIS/VIIRS Subsets' with a description: 'Moderate Resolution Imaging Spectroradiometer / Visible Infrared Imaging Radiometer Suite Land Products Subsets'. On the right, there's a 'Feedback' link and a question mark icon. Below the banner is a dark blue navigation bar with links: 'Home', 'Get Data', 'Documentation', 'Resources', 'Publications', 'Citation', and a 'Sign in' button. Below the navigation bar, there's a breadcrumb trail: 'Home > Get Data > Global Subsets Tool'. The main heading is 'Global Subsets Tool: MODIS/VIIRS Land Products'.

You'll need to register by clicking on the **"Sign in"** button and following instructions to specify a valid password. After doing this successfully, the button will change to **"Sign out"**.

Username ?

don.mcneil

Password

.....

☒ Stay signed in (this is a private workstation)




LOG IN


REGISTER

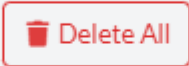
To download LST data, you need to select the product MOD11A2 instead of the default (MOD13Q1), and then click on the “**Delete All**” button to change the default location (Oak Ridge in Tennessee) to the centre of the data pixel.

The default subset size is 3 x 3, which corresponds to 49 pixels in a 7 by 7 array with approximate area 42 km².

ID: 1
Point
Location: 35.9625°, -84.295822°
Subset Size: 3 km (N-S) x 3 km (E-W)



 Add Location

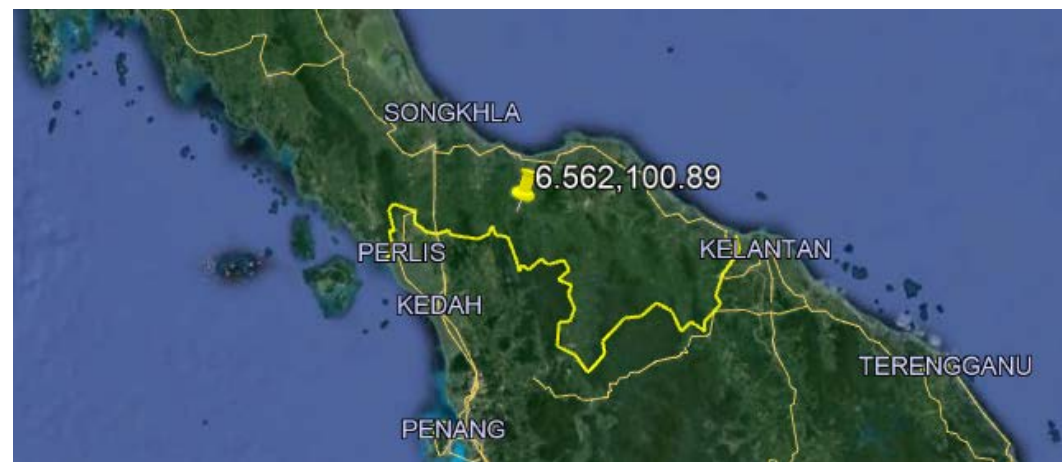
 Delete All

2. Select the Product(s): 1 of 34 products selected

MOD11A2 x

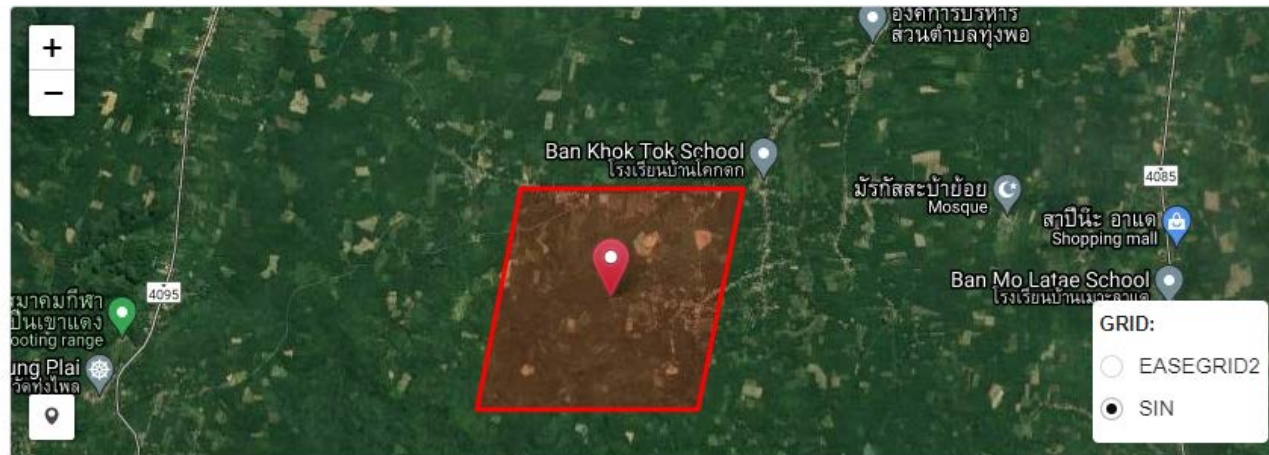
<div>MCD15A3H ⓘ Leaf Area Index (LAI) and FPAR (Terra + Aqua), 4-Day, 500m</div> <div>+ Select</div>	<div>VNP15A2H ⓘ Leaf Area Index (LAI) and FPAR (S-NPP), 8-Day, 500m</div> <div>+ Select</div>	<div>MYD15A2H ⓘ Leaf Area Index (LAI) and FPAR (Aqua), 8-Day, 500m</div> <div>+ Select</div>	<div>MOD11A2 ⓘ Land Surface Temperature and Emissivity (Terra), 8-Day, 1000m</div> <div>x Unselect</div>
---	--	---	---

If you specify latitude 6.562 and longitude 100.89 and click on **"Add Location"** the picture below will appear showing the area covered by the 49 pixels.



Global Subsets Tool: MODIS/VIIRS Land Products

1. Specify the Location(s): 1 of maximum 30 locations



ID: 1
Point
Location: 6.563°, 100.89°
Subset Size: 3 km (N-S) x 3 km (E-W)



+ Add Location

Delete All

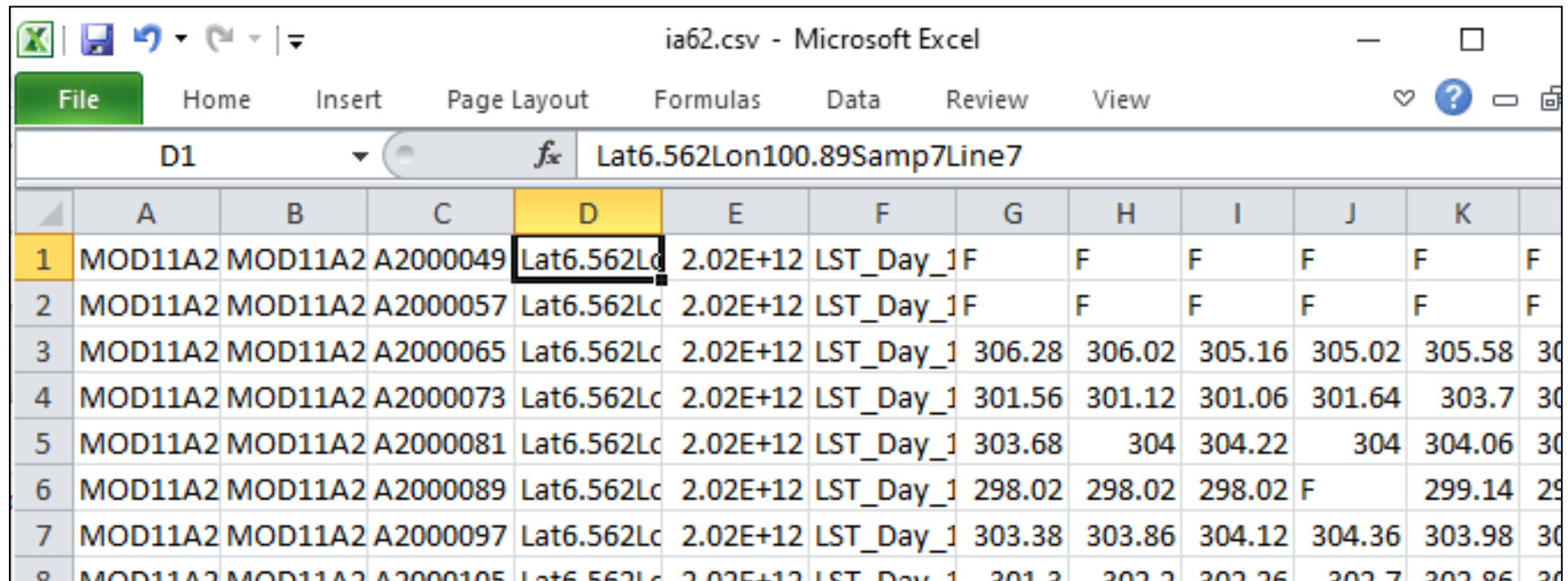
Finally, scroll down and click on the **"Submit Order"** button to place your order.

When the order arrives, click on the file name and then **"filtered_scaled_LST_Day_1km.csv"** to get daytime LST data as a CSV file.

The data are stored as a CSV text file in a table with 55 columns and as many rows as there are 8-day periods from day 49 in year 2000 until the latest date for which data from the satellite have been processed.

Column 3 specifies the dates of successive measurements and column 4 contains the latitude and longitude in each row.

Columns 7 to 55 specify mean Land Surface Temperatures in degrees Kelvin within each pixel at west-to-east longitudes and north-to-south latitude bands.



	A	B	C	D	E	F	G	H	I	J	K
1	MOD11A2	MOD11A2	A2000049	Lat6.562Lon	2.02E+12	LST_Day_1	F	F	F	F	F
2	MOD11A2	MOD11A2	A2000057	Lat6.562Lon	2.02E+12	LST_Day_1	F	F	F	F	F
3	MOD11A2	MOD11A2	A2000065	Lat6.562Lon	2.02E+12	LST_Day_1	306.28	306.02	305.16	305.02	305.58
4	MOD11A2	MOD11A2	A2000073	Lat6.562Lon	2.02E+12	LST_Day_1	301.56	301.12	301.06	301.64	303.7
5	MOD11A2	MOD11A2	A2000081	Lat6.562Lon	2.02E+12	LST_Day_1	303.68	304	304.22	304	304.06
6	MOD11A2	MOD11A2	A2000089	Lat6.562Lon	2.02E+12	LST_Day_1	298.02	298.02	298.02	F	299.14
7	MOD11A2	MOD11A2	A2000097	Lat6.562Lon	2.02E+12	LST_Day_1	303.38	303.86	304.12	304.36	303.98
8	MOD11A2	MOD11A2	A2000105	Lat6.562Lon	2.02E+12	LST_Day_1	301.2	302.2	302.26	302.7	302.86

This is just one sample of data from a much larger population.

This population covers the land area of the whole world, which an Internet search tells us is just under 149 million square kilometers.

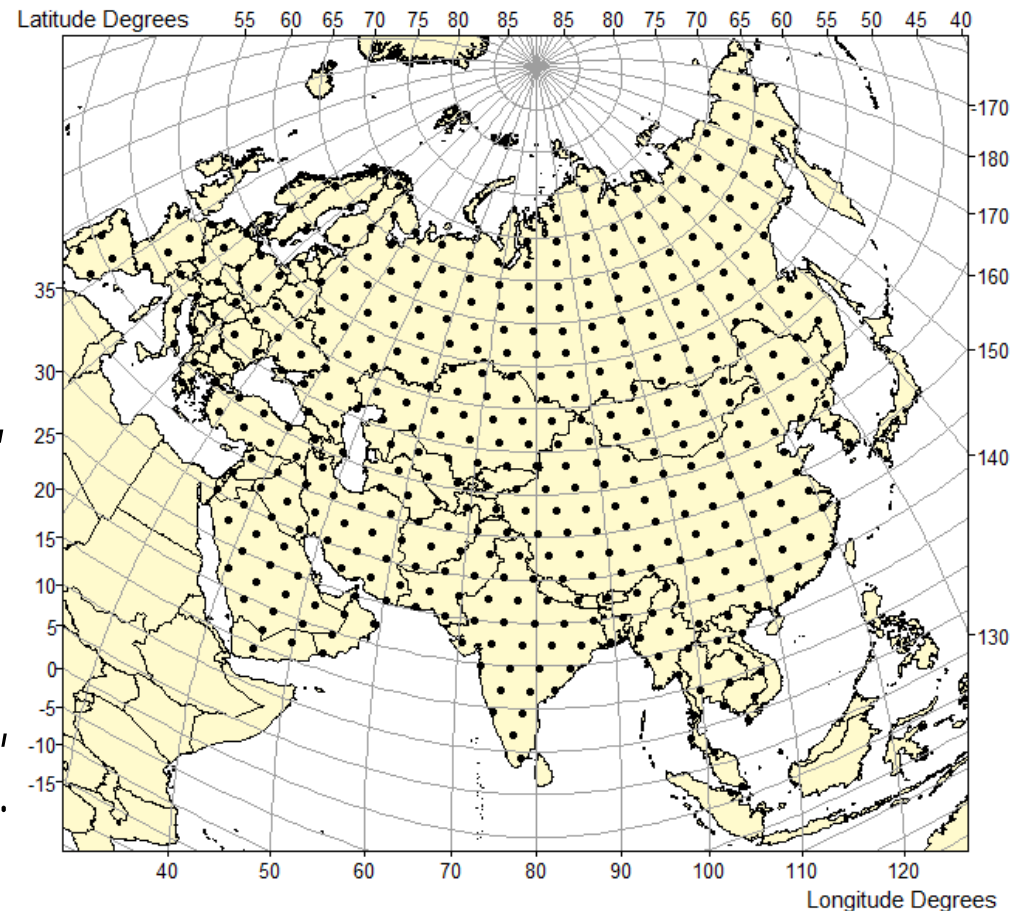
Samples of area 42 km^2 give population size $149,000,000/42 = 3,547$ million.

But to find out how LST has changed over the whole planet, we don't need to analyze all these data. Opinion pollsters survey large populations accurately using sample sizes of 1000 or less.

An unbiased sample should cover the whole population, ensuring that all different components are included.

We could do this by making a regular grid of points over the Earth's surface, spacing sample points equally around latitude bands, themselves spaced at constant distances apart.

This Europe/Asia map has 450 points, each representing similar-sized areas.

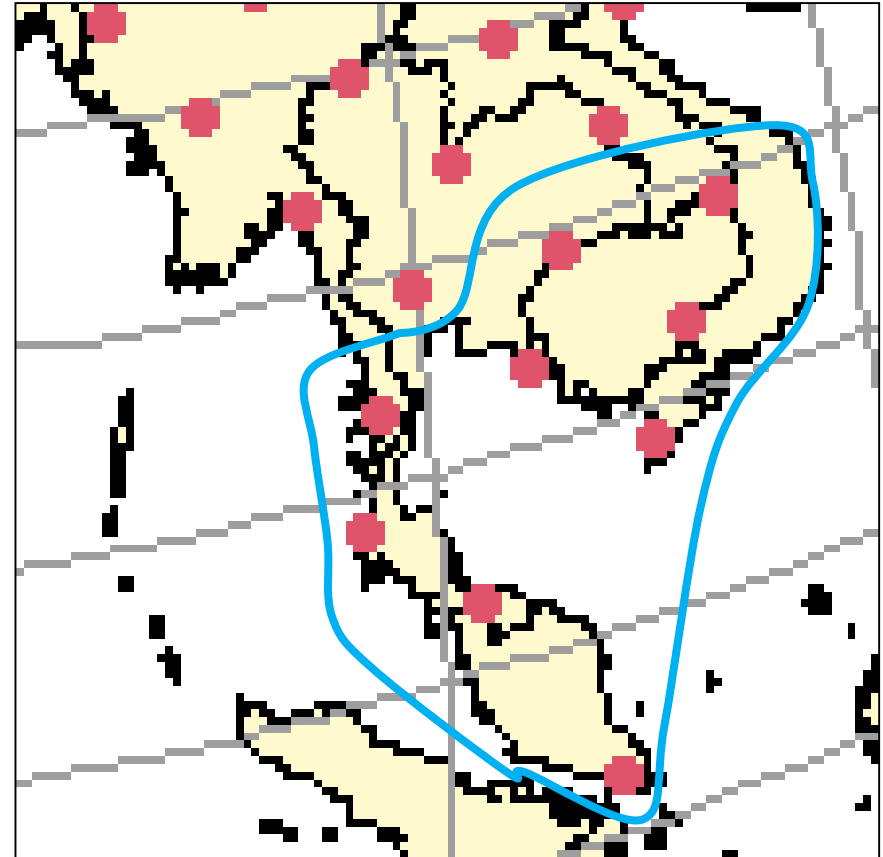


The grid map for Europe and Asia can be divided into 50 contiguous *regions* each containing a sub-sample of nine *sub-regions*, where sub-regions are defined as arrays containing 49 pixels similar to that shown on Slide 7.

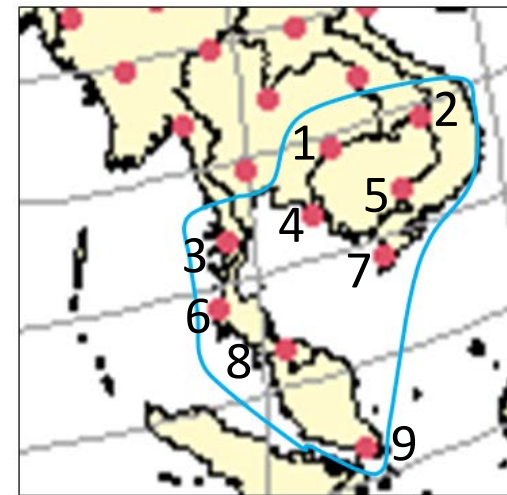
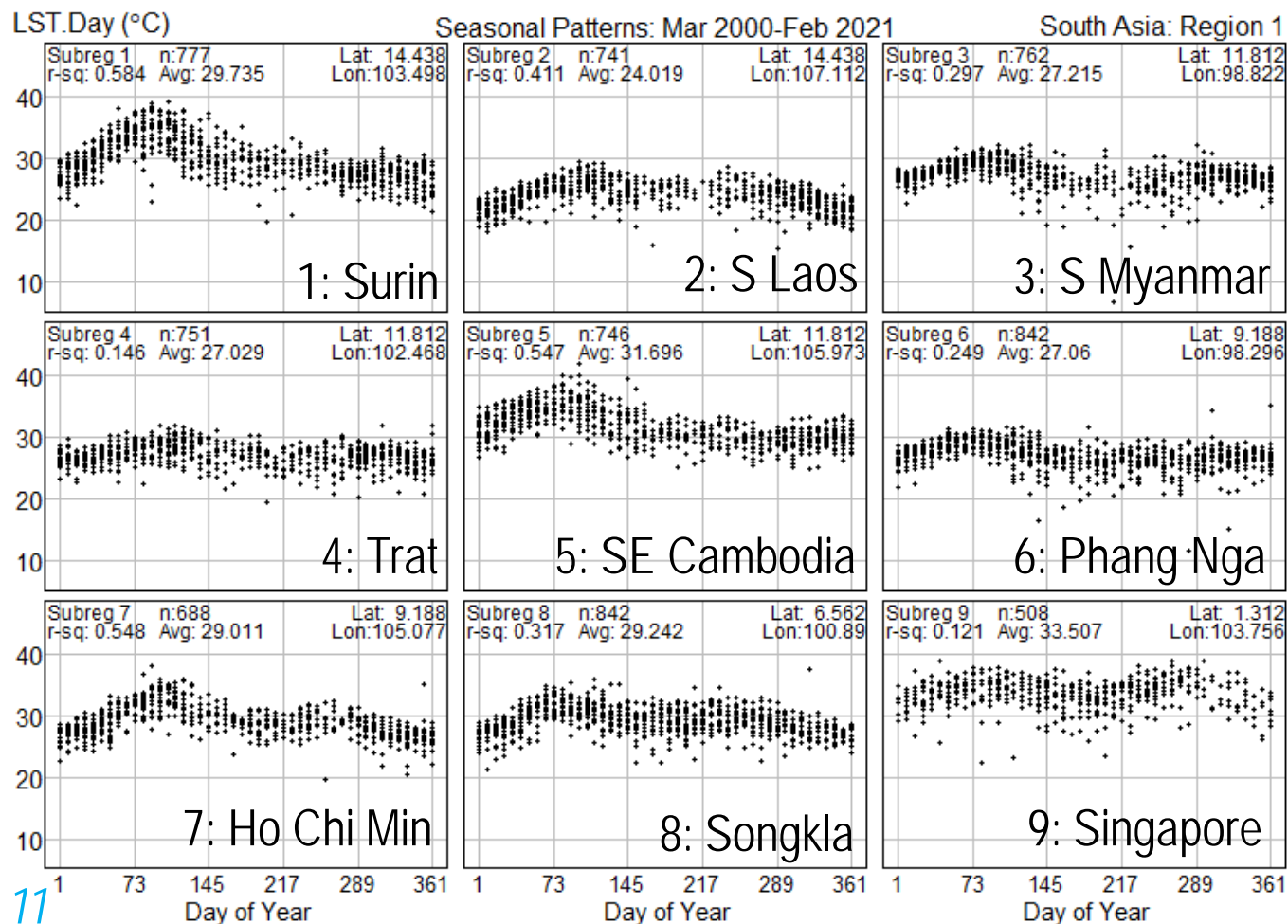
The map on the right shows a blue-ringed region containing nine sub-regions in Thailand, Laos, Myanmar, Cambodia, Thailand, Vietnam, West Malaysia and Singapore.

We'll use LST data in this sample to illustrate data analytic methods.

This will involve graphing the seasonal patterns, fitting models to these curves, seasonally-adjusting them to create time series of temperature trends, and fitting further models to compare and forecast global warming trends.

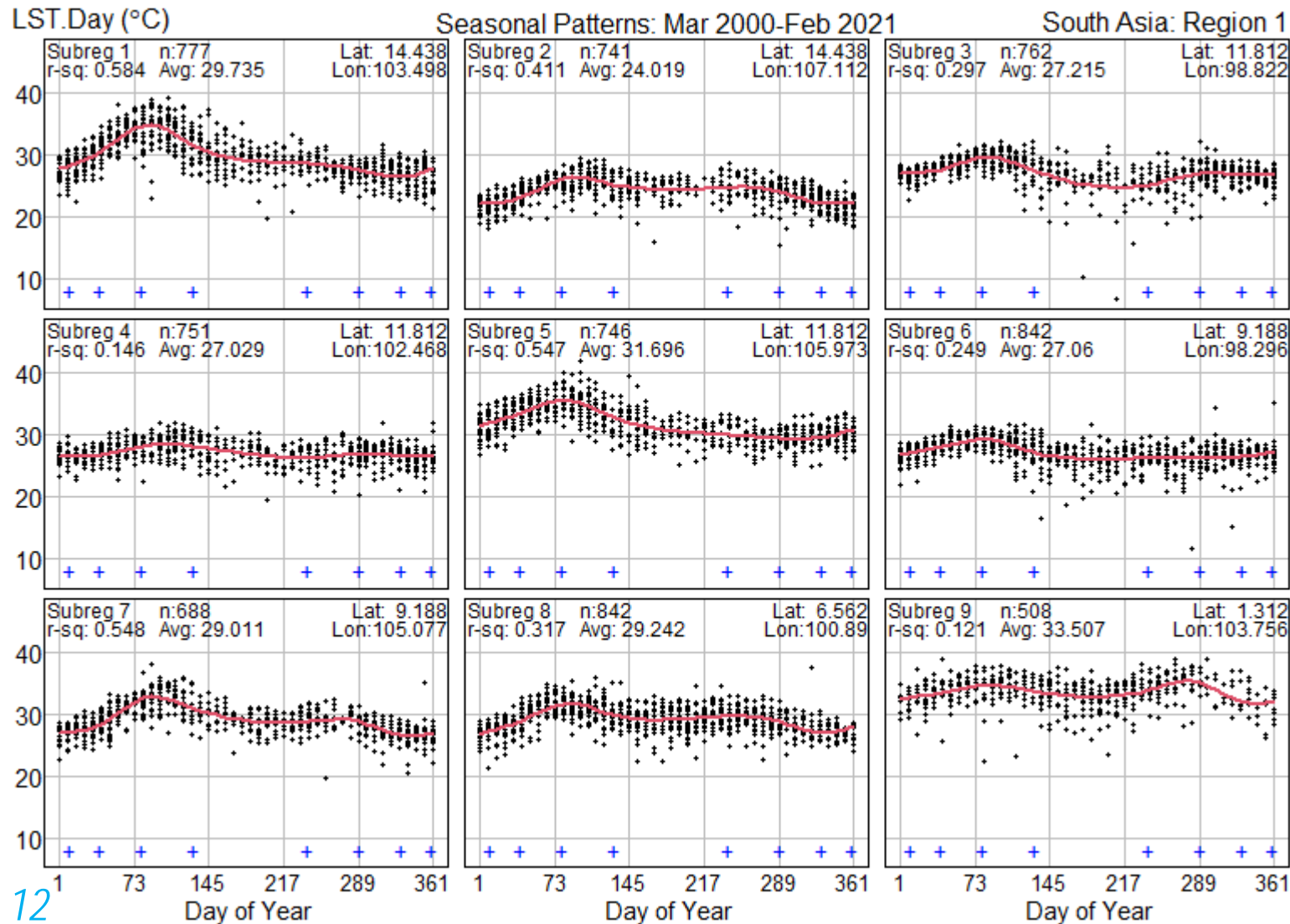


These graphs show seasonal patterns for the nine sub-regions. You can create them by executing the code up to `#-----pause1` in the R program `iaTD5.Rcm` with CSV data files `ia53-ia54`, `ia56-ia58`, and `ia60-ia63`.



Time gaps between observations are 8 days except after day 361 (5 days or 6 days in a leap year).

Red curves denote natural cubic spline functions with eight knots (shown as the blue plus-signs) fitted to the seasonal patterns. A boundary condition is needed to make slopes match at end-points. To create this graph, insert a # (comment) symbol before the statement on line 3 (**showSpline=="no"**).



Extensive data testing on LST data suggests that eight knots are sufficient, but fewer knots are needed in the middle range where most missing values occur.

Linear Regression Model to fit Seasonal Pattern using Natural Spline

$$y = a + bx + \sum_{k=1}^{p-3} c_k s_k(x) \quad \text{Formula (linear function of } a, b, c_1, c_2, \dots, c_{p-3})$$

$$\text{where } s_k(x) = (x-x_k)_+^3 - \frac{(x_p-x_k)(x_{p-1}-x_k)}{d_3 d_2} (x-x_{p-2})_+^3 + \frac{(x_p-x_k)(x_{p-2}-x_k)}{d_1 d_2} (x-x_{p-1})_+^3 - \frac{(x_{p-2}-x_k)(x_{p-1}-x_k)}{d_3 d_1} (x-x_p)_+^3,$$

(three boundary conditions) $x_+ = \max(x, 0)$, and $d_1 = x_p - x_{p-1}$, $d_2 = x_{p-1} - x_{p-2}$, $d_3 = x_p - x_{p-2}$.

Computer Program # LST values for subregion j

```

yy <- as.data.frame(yt[,j+1])
names(yy) <- "y1"
yy$x <- as.integer(doy) # day of year
x <- yy$x
kn <- c(10,40,80,130,240,290,330,360) # spline knot locations
p <- length(kn)
d1 <- kn[p]-kn[p-1]; d2 <- kn[p-1]-kn[p-2]; d3 <- kn[p]-kn[p-2]
for (k in c(1:(p-3))) {
  sk <- ifelse(x>kn[k],(x-kn[k])^3,0)
  sk <- sk-((kn[p]-kn[k])*(kn[p-1]-kn[k])/(d3*d2))*ifelse(x>kn[p-2],(x-kn[p-2])^3,0)
  sk <- sk+((kn[p]-kn[k])*(kn[p-2]-kn[k])/(d1*d2))*ifelse(x>kn[p-1],(x-kn[p-1])^3,0)
  sk <- sk-((kn[p-2]-kn[k])*(kn[p-1]-kn[k])/(d3*d1))*ifelse(x>kn[p],(x-kn[p])^3,0)
  yy[, (k+2)] <- sk
  names(yy)[k+2] <- paste("s",k,sep="")
}
mod1 <- lm(data=yy,y1~x+s1+s2+s3+s4+s5) # fit linear model for p=8
    
```


Choice of Number of Spline Knots

The knots cover the range of the data, so extreme knots correspond to the minimum and maximum time points.

We always use *natural* splines, defined as piecewise cubic functions that are linear beyond the range of the data. With two boundary conditions needed to make the function linear outside the data range, the formula is as follows.

$$y = a + bx + \sum_{k=1}^{p-2} c_k s_k(x)$$

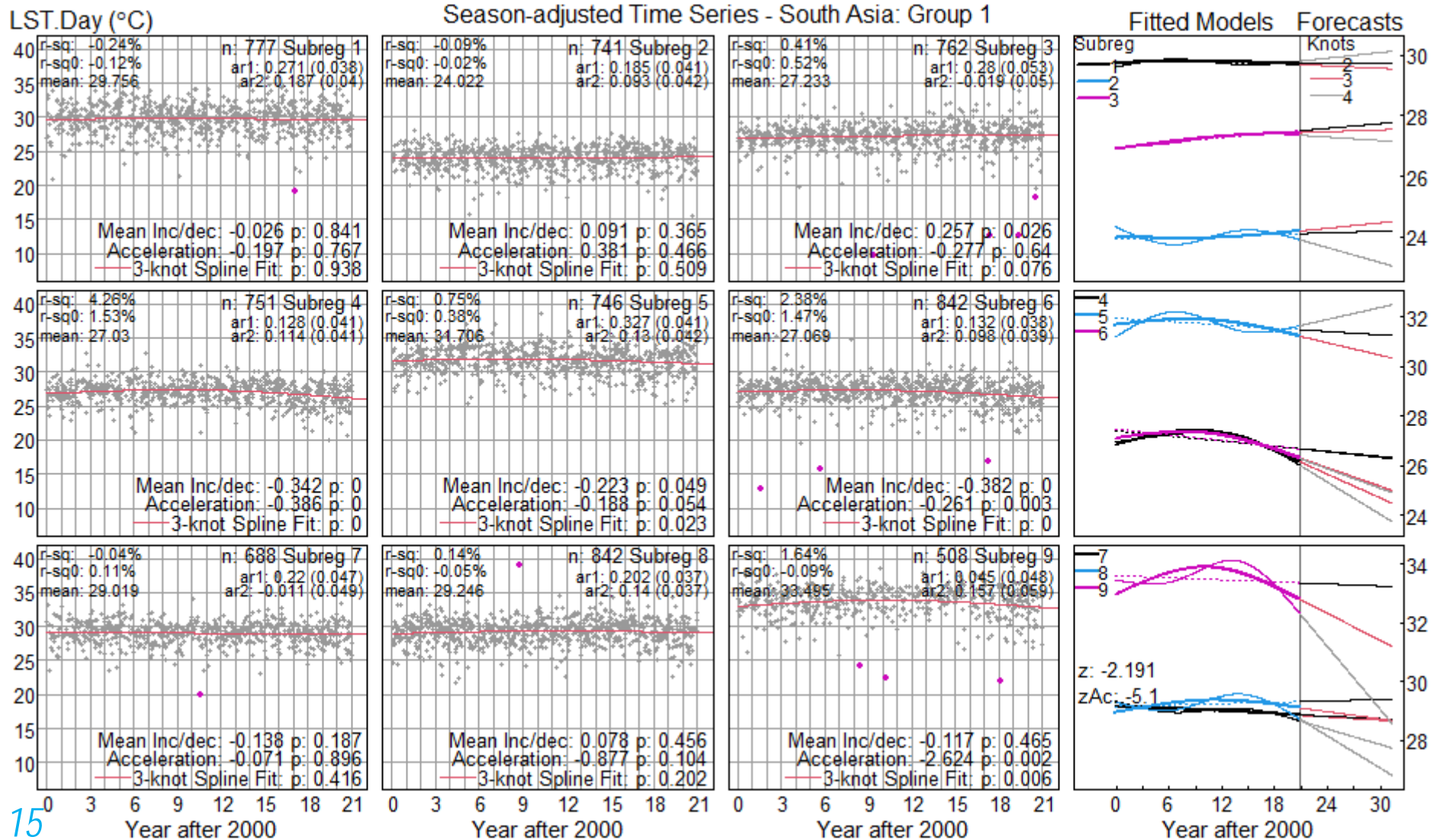
$$\text{where } s_k(x) = (x - x_k)_+^3 - \frac{(x_p - x_k)}{d_1} (x - x_{p-1})_+^3 + \frac{(x_{p-1} - x_k)}{d_1} (x - x_p)_+^3 \text{ and } d_1 = x_p - x_{p-1}.$$

With only two knots, the formula is $y = a + bx$, just a straight line.

With three knots, the spline has three parameters, like a quadratic, but this function is more useful in practice because its forecasts are linear, whereas forecasts based on quadratics tend to overshoot or undershoot data.

With more than three knots, splines can detect periodic waves in data, and might provide better short-term forecasts, but are less accurate for long-term forecasts because they tend to increase or decrease too rapidly, as the next slide shows.

Red curves here denote natural cubic spline functions with three equispaced knots at years 0, 10.5 and 21 fitted to season-adjusted time series patterns. Fitted splines on the right have 2, 3 and 4 knots and corresponding forecasts. To create this graph, run the remaining lines in the [iaTD5.Rcm](#) program.



Prediction after using the lm() Function in R

Suppose that a data table **yy** contains a set of columns **xx** containing predictor variables for an outcome of interest and another column **y1** containing corresponding values of the outcome variable. The R program listed on Slide 13 gives an example where **xx** contains five predictors.

The command **mod1 <- lm(data=yy,y1~x+s1+s2+s3+s4+s5)** fits a linear regression model and stores results in **mod1**, including fitted values that you can summarise using **summary(mod1\$fit)**. However, if there are missing values in the outcome these values are also absent from the list of fitted values, even though regression models are designed to predict them.

Climate data reported by NASA satellites have lots of missing values, and even small proportions of missing data can substantially distort results in multivariate analysis, so a method is needed to impute them.

The **predict()** function in R can do this, simply by using a command such as **fv <- predict(mod1,yy)** instead of **fv <- mod1\$fit**. And it can also be used to create forecasts when analyzing time series data, such as those shown in the panels on the right of the graph on Slide 15 where the fitted spline functions are extended by 10.5 years into the future.

Acceleration

The results shown in the plots on Slide 15 show substantial variation among the nine sub-regions sampled. Three sub-regions show statistically significant decreases in daytime LST, one shows a statistically significant increase, and the other six show no conclusive evidence of change over the 21-year period. While this sample is too small to allow us to make global conclusions, the method can be applied to all sub-regions in a grid covering the whole planet, and we'll do this next. But in doing so, other questions arise, and more extensive methods are needed.

For example, we could ask if LST increase is accelerating or decelerating. This question can be addressed by fitting splines with three knots and assessing whether the slope of the spline at the end of the observation period is greater or less than its initial slope. The answers to this question for our sample of nine sub-regions are seen in the numbers associated with the **Acceleration:** legend that shows the increase in the estimated LST increase per decade over the 21 year period and its p-value for testing the null hypothesis of no change. We see that three sub-regions decelerated and the remaining six showed no conclusive evidence of change.

Time Series Correlation

Regression models require statistical assumptions including independence of errors. For time series data where a model separates the data into a signal containing the available information and a series of residual errors, these errors may be correlated, in which case this autocorrelation structure also needs to be addressed.

The `arima()` function in R can be used to fit a regression model that takes such autocorrelation structure into account. For example, to fit a model with two auto regression parameters with predictor variables in `xxj`, the statement

```
zj1 <- arima(yyj,order=c(2,0,0),xreg=xxj)
```

may be used, where `xxj` contains the terms in a spline function with two, three or four knots as defined on Slide 14. For two knots `xxj` is just `x`, for three knots it comprises `x` and `s1(x)`, and for three knots it contains the three columns `x`, `s1(x)` and `s2(x)`. For the two-knot model, estimates for the auto regression parameters with their standard errors bracketed are shown on Slide 15 as `ar1` and `ar2`.

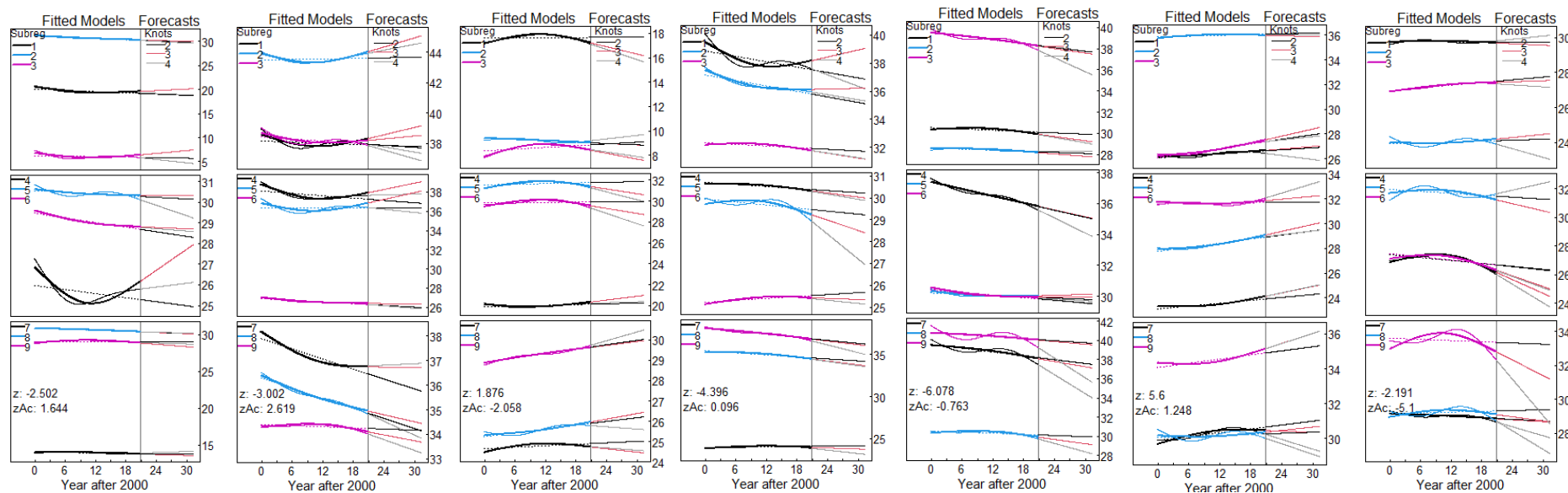
Multivariate Regression

So far we have focused on fitting regression models to average values of land surface temperatures within a specified sub-region. But if we want to know what is happening in a larger region such as the whole area within the blue ring shown on Slide 11, the results from the nine sampled sub-regions need to be aggregated in some way. Computing the average of the nine component estimates provides an answer, but unless these components are mutually independent, its standard error is not easily calculated.

It turns out that most climate variables are spatially correlated, even when hundreds of kilometers apart. However, correlated multivariate outcomes in regression models can be handled straightforwardly using **multivariate linear regression**, and the **lm()** function in R does this by specifying the outcome variable as a matrix rather than a single column. For example, the **iaTD5.Rcm** program uses the statement **lm(as.matrix(ySa)~ySA\$t[1:nObs]) -> mod** to do this, where **ySa** contains the nine season-adjusted LST variables (also filtered to remove time series autocorrelation using the **arima()** function).

We use this method to analyze LST increase and acceleration within regions. Standard errors for average LST increases in regions are then computed from a variance-covariance matrix (an attribute of **mod**).

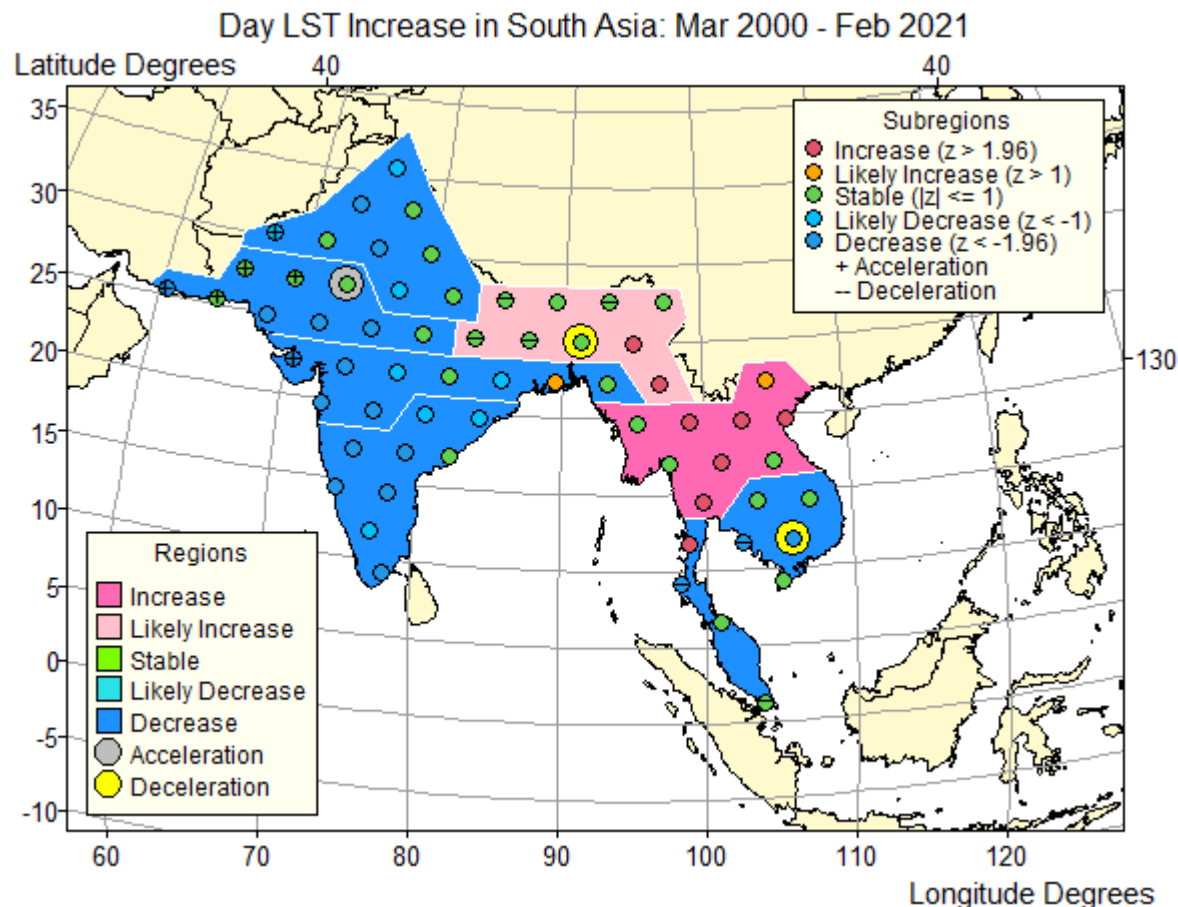
These graphs show three fitted models for each of nine sub-regions in seven regions of the south Asian continent, and forecasts up to 10.5 years ahead. z values are based on statistical tests of the null hypothesis that there is no overall daytime LST increase in a region, assuming that increases in sub-regions of a region are the same. Decreases occurred in five regions, one region increase, and one region showed a likely increase (z between 1 & 1.96). zAc values result from similar tests on acceleration. Two regions decelerated and one region accelerated.



This map shows daytime LST increases for seven regions in the south of continental Asia, including Pakistan, India, Nepal & Bhutan, Bangladesh and seven ASEAN countries.

With only 63 sub-regions in the sample, it is not large enough to make firm conclusions.

However, the sample size could be increased by reducing the distances between sub-regions, and it will be interesting see corresponding results.



To summarize, we have applied basic data analytic methods to a sample of daytime land surface temperature remote sensing data reported from Earth-orbiting satellites from March 2000 to February 2021. The methods involve relatively simple regression models and don't require complex programs. They just use the base R software system, which is freely available and open source, and don't require any additional libraries.

Our sample is too small and possibly too short in duration to say conclusively how rapid or widespread global warming is. But with increased sample sizes and further time, results will become clearer. Within a few months, we'll have another year of data.

Next week we'll extend the sample to include the whole of Europe and Asia, and other continents and islands around the world.

Please email me at don.mcneil@mq.edu.au if you'd like to work with us on this important research topic.

Thank you for your patience. Hope to see you next week!

Data Analytic and Empirical Forecasting Methods using Smart Linear Regression: Session 2

Don McNeil

Emeritus Professor, Macquarie University, Australia

Prince of Songkla University, Thailand, 30 January 2022

National Aeronautics & Space Administration (NASA) Data

Recap of Session 1

Increasing the Sample Size

New Results for an ASEAN Region

Confidence Intervals and Forest Plots

Creating a Thematic Map

Using a Sinusoidal Projection

Graphs of Models with Forecasts

Take-home Message

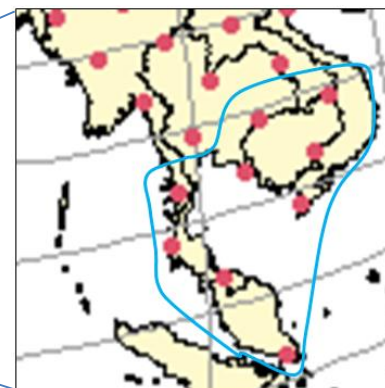
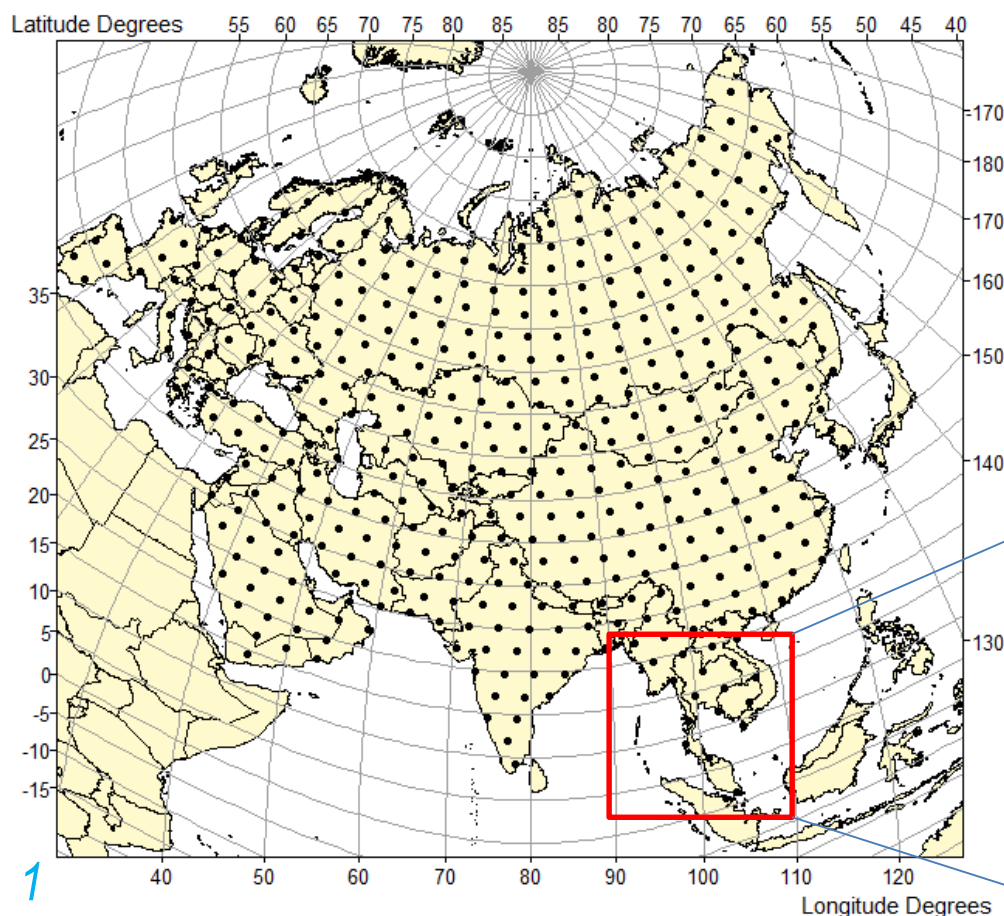


Last week we applied basic data analytic methods to land surface temperature (LST) data from Earth-orbiting satellites recorded from March 2000 to February 2021 at 8-day intervals from sub-regions each covering 7×7 pixel arrays (area 42 km^2) downloaded from a NASA website. The sample comprised just nine sub-regions taken from a regular grid of 450 similar sub-regions covering the

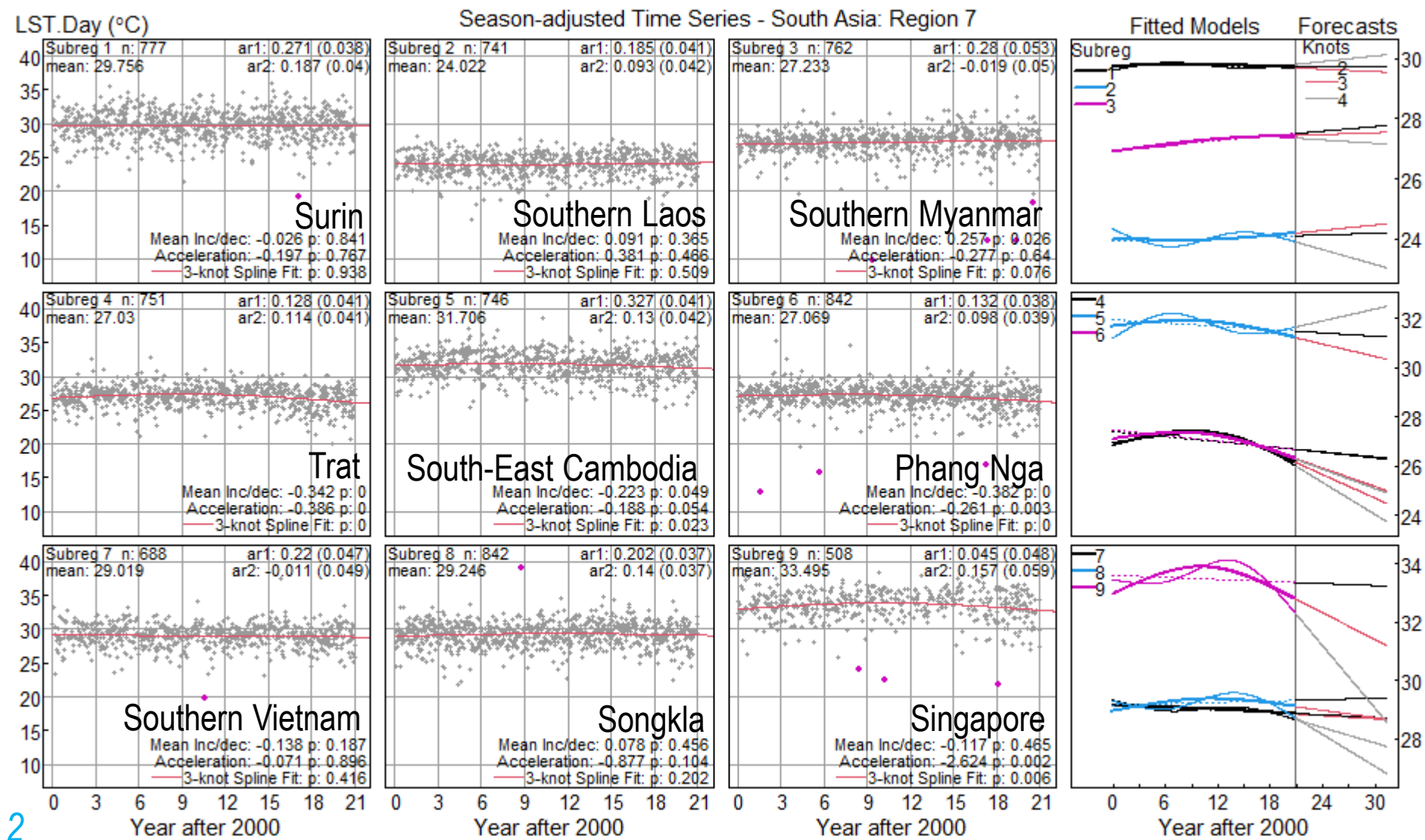
Europe/Asia continental land mass.

Using a *linear regression model* to estimate LST increase in this region, the *z-value* was -2.191, indicating a statistically significant **decrease** in day land surface temperature.

But is a sample of **nine** large enough to be conclusive?



This graph shows time series plots of season-adjusted day LST in the nine sub-regions as shown on Slide 15 from last week's session. R-squared statistics are of minor relevance so are not shown. P-values are much more important.



The size of a sample does not depend on the population size. Provided that the sample is representative of the population and includes its different components in proportion to their presence in the population, a sample just needs to be large enough to give sufficiently precise estimates.

It's quite possible that a sample of nine is large enough to provide a precise estimate. This is true when all members of a population are very similar, like grains of sand on a beach. But in other situations, such as stars in the night sky that include dwarfs and giants, where there is large variation between different members of the population, the sample size needs to be much larger.

The grid we have used for the Europe/Asia continent uses distances of 420 pixel widths (360.0 km) between successive points around latitude bands and 315 pixel widths (270.0 km) between successive points around longitude great circles. These numbers are divisible by 3, 5 and 7, making it easy to pack sub-regions more tightly within a study area of interest. Let's see what happens when these gaps are reduced to 140 and 105 pixel widths, respectively.

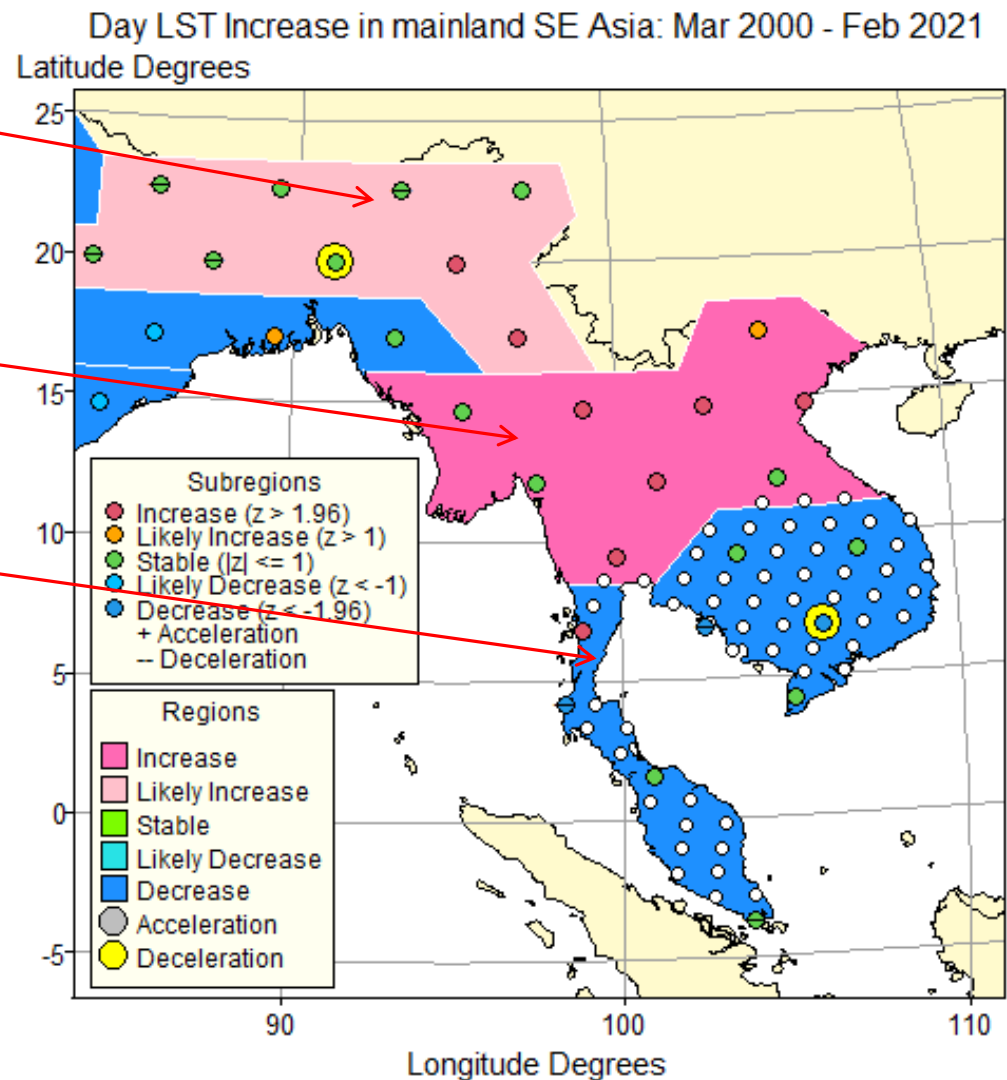
This map shows how daytime land surface temperature increases varied in three eastern regions of the south Asian group, as shown on Slide 21 from last week's session. Results for the two easternmost regions differ substantially.

In east India & northern Myanmar, day LST “probably” increased.

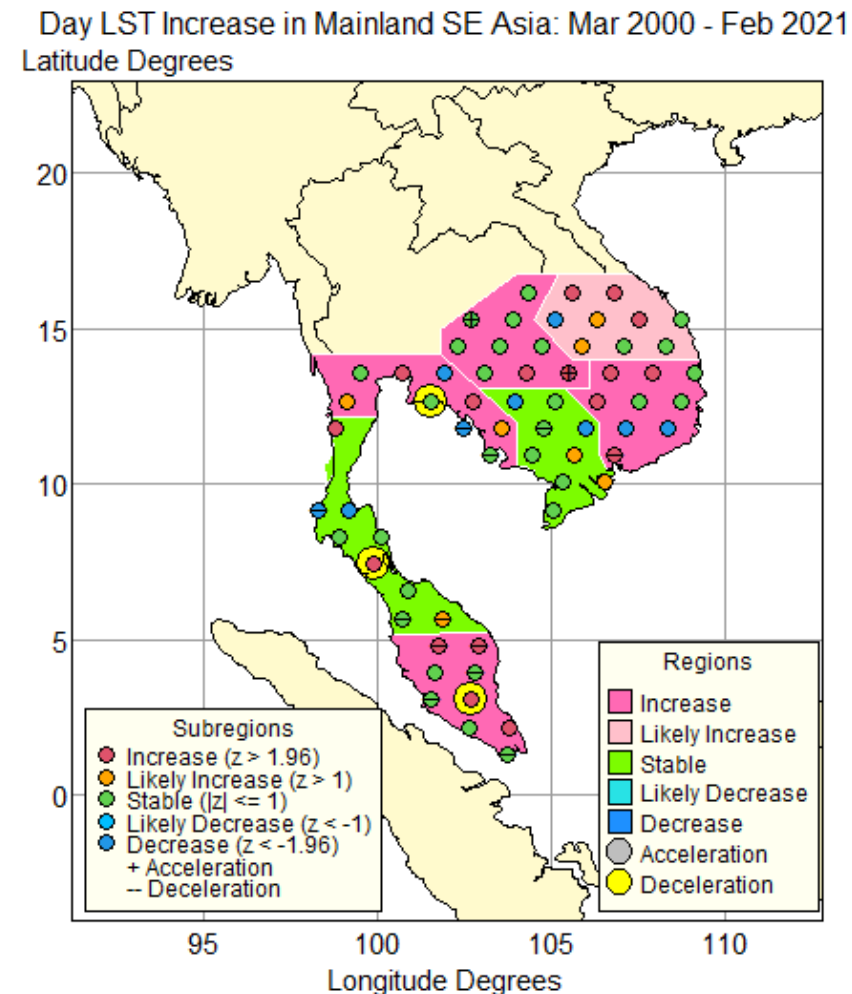
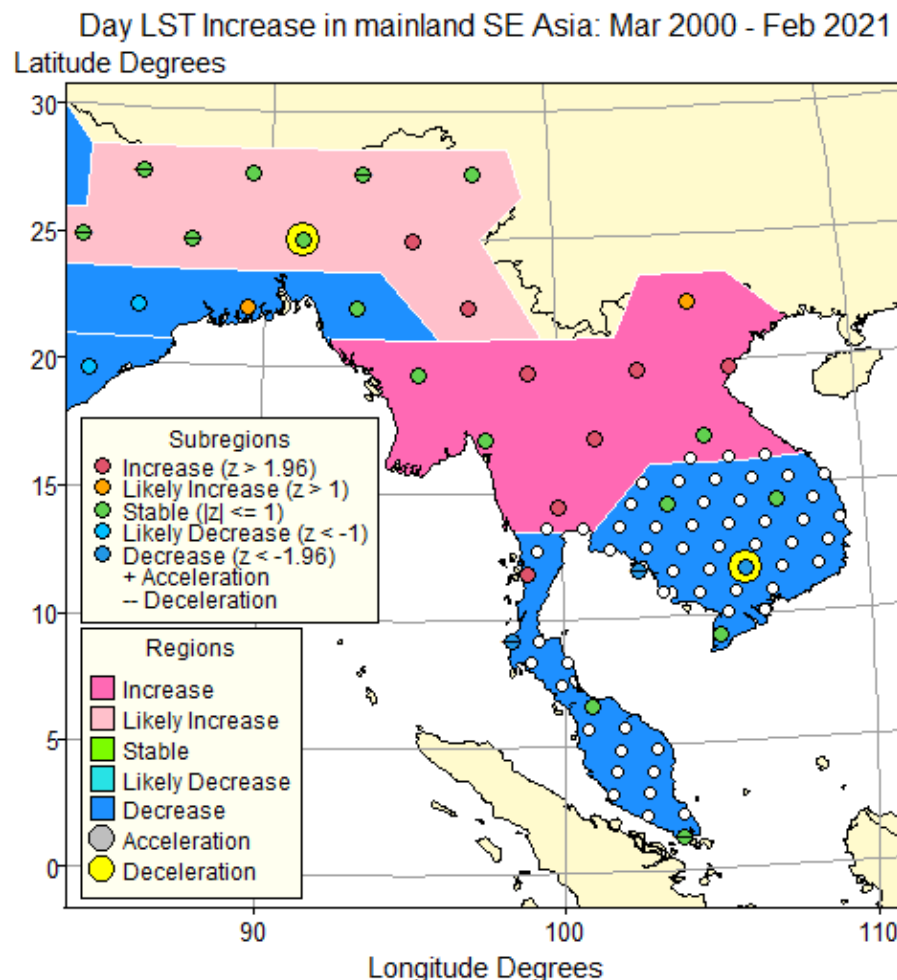
In Bangladesh, northern areas of Thailand, Laos, Thailand and Vietnam, day LST **increased**.

But in the south of this area, day LST **decreased**.

The white dots show a denser grid that expands the number of sub-regions in this southern region from 9 to 63 by inserting two additional grid points between neighbouring pairs.



The results for the south-eastern region in the south Asian group with the sample size increased to 63 (right panel) are quite different to those for the same region with sample size nine (left panel). How can this be explained?

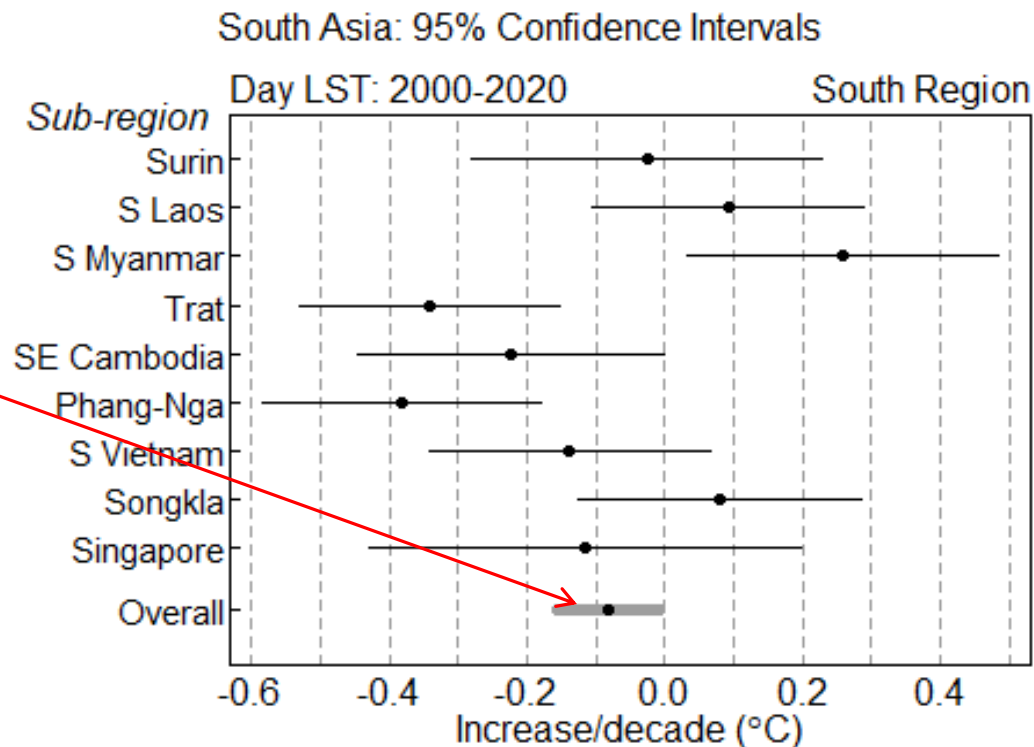


Confidence intervals (CIs) provide an explanation. The “smart” regression model we’re using to estimate LST increases also provides estimates of **standard errors**, which in turn tell us how accurate these results are.

A **95% confidence interval** is an interval that contains an unknown population parameter with probability 0.95. If model assumptions are satisfied, to a close approximation these intervals have width twice the standard error on each side of the model estimate.

Here’s a graph (known as a **forest plot**) of 95% CIs for LST increases in the nine sub-regions, with an overall 95% CI obtained by aggregating all nine samples.

This overall CI is based on a **homogeneity assumption** that the population parameter is the same in each sub-region.

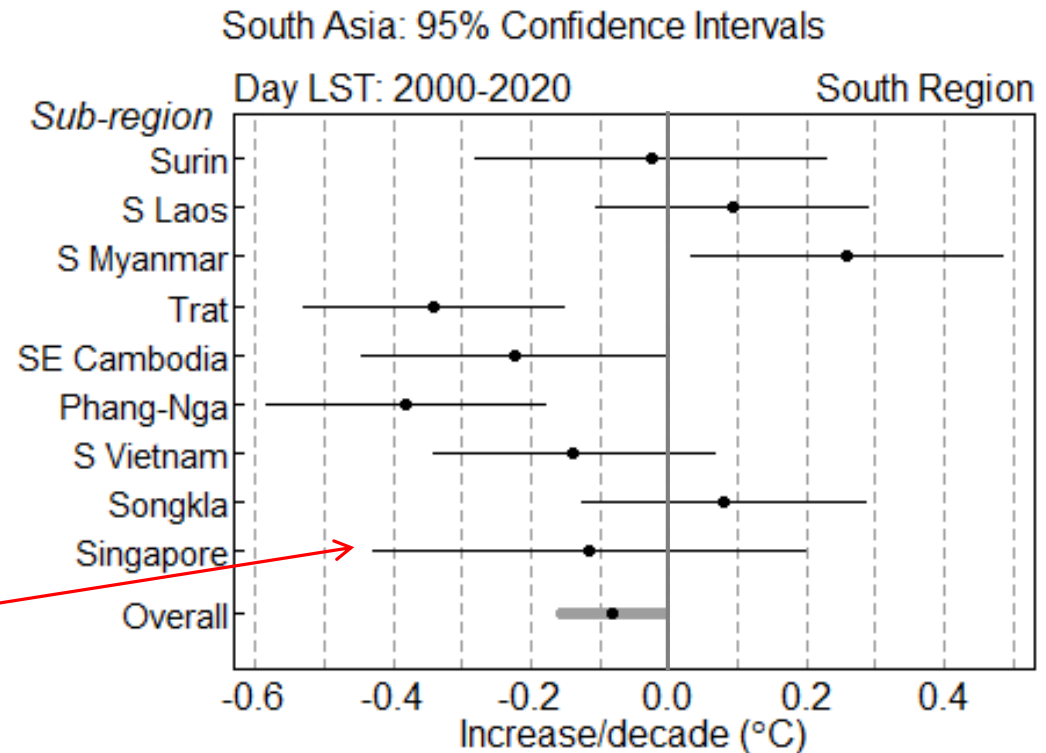


An important rule of thumb in statistics is that the standard error (**SE**) of an estimate decreases in proportion to the square root of the sample size (n).

This leads to the simple formula $SE = sd/\sqrt{n}$, where **sd** is the standard deviation of a single population member

So for a sample of nine, this rule suggests that a CI for the overall average should be one-third the length of that for a single component, as our forest plot confirms, except for Singapore.

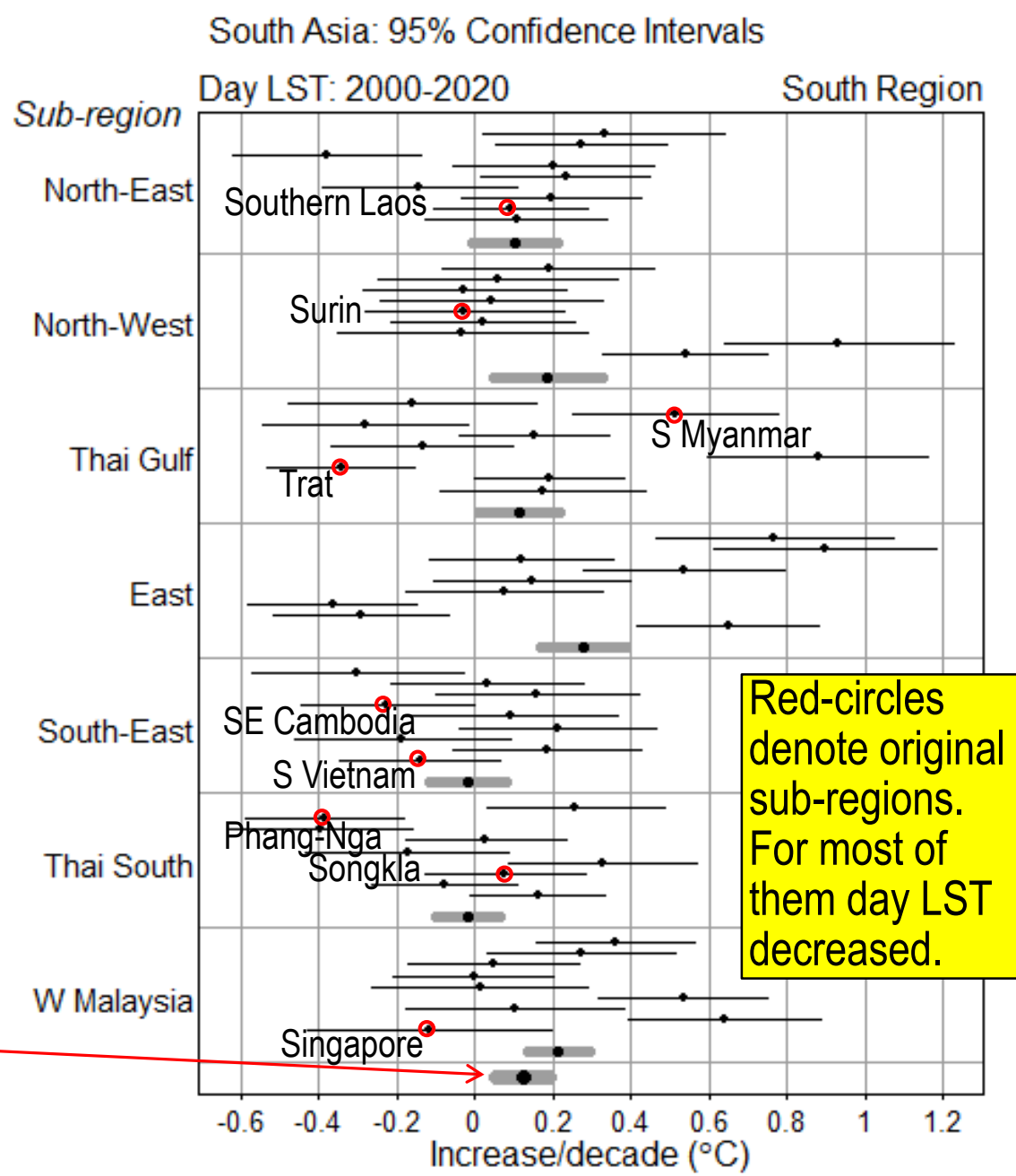
But nearly half of the data for Singapore are missing, so its confidence interval is wider.



So the homogeneity assumption is highly questionable for these data. The upper three sub-regions on the plot have higher day LST increases than the middle three, and this fact strongly suggests that the result is wrong.

When the number of sub-regions in the study area is increased to 63, the forest plot is more informative.

Confidence intervals are again shown for sub-regions in each of the smaller regions together with their overall CI, as well as a CI based on aggregating all 63 sub-regions. If we can now assume homogeneity of these increases (and this remains questionable), for the study area as a whole we see that day LST **increased**, in contrast to the original conclusion.



The program that creates graphs for regions similar to that shown on Slide 2 (**saTD5b.Rcm**) also stores results in text files that can in turn be used to create thematic maps similar to those shown on Slide 5. The map on the left of this slide uses a sinusoidal polar projection that matches the nearly circular orbit of the Terra satellite. To create this map we need to use this projection. The method is described later in this session.

The map on the right of Slide 5 is simpler to create because it just uses longitude and latitude as Cartesian coordinates. The program is in the file **seAsia.Rcm**. It uses shape files for different countries stored as four variables named **plotID**, **pointID**, **x** and **y** in a database table. To create this map, the countries needed are Vietnam (plotID: 236), Cambodia (plotID 36), Thailand (plotID 213), West Malaysia (155.4), Myanmar (plotID 25), with some additional countries and islands. These data are stored in the file **wsea.csv**. The file **saLSTinc.txt** is created or updated when **saTD5b.Rcm** is executed.

To test your programming skills, see if you can create the map below.

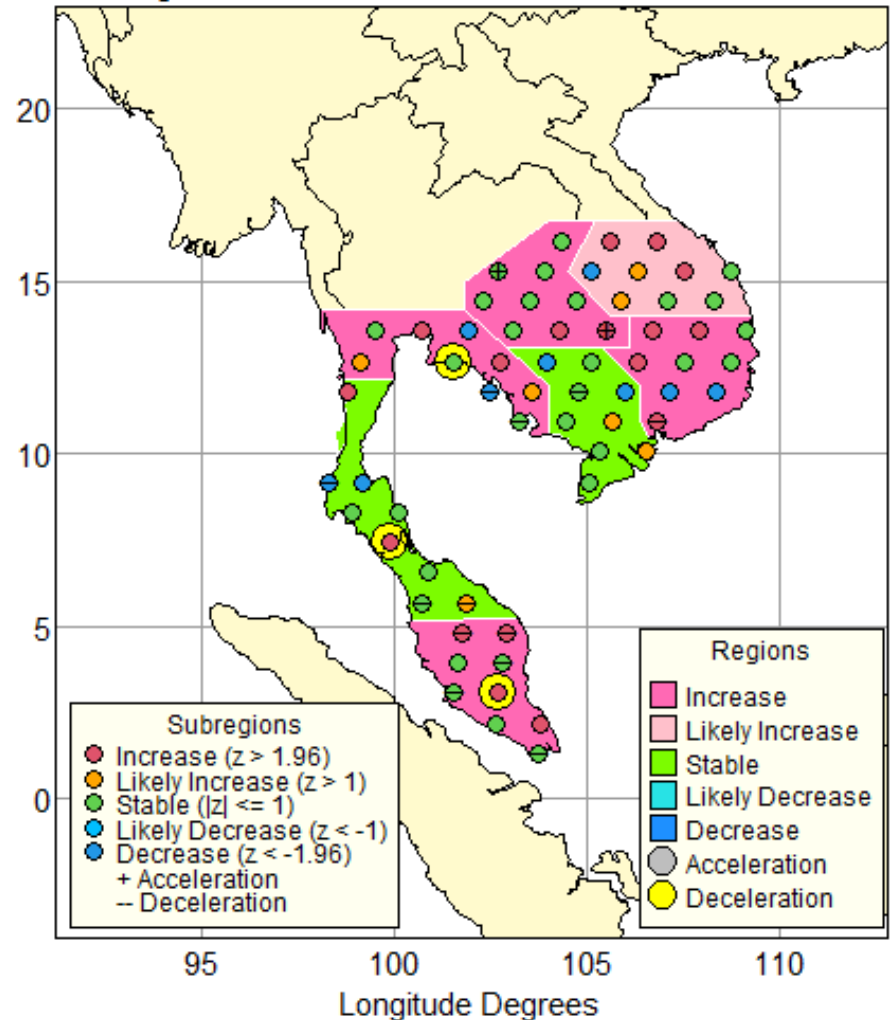
The steps are as follows.

1: Copy the program file **seAsia.Rcm** and data files **wsea.csv** and **saLSTinc.txt** into your working directory. If the name of this directory is not **c:/world**, edit **seAsia.Rcm** by changing the statement **setwd("c:/world")** to contain this folder name.

2: Open R and copy the contents of the file **seAsia.Rcm** into its command window. If the map appears, take a bow.

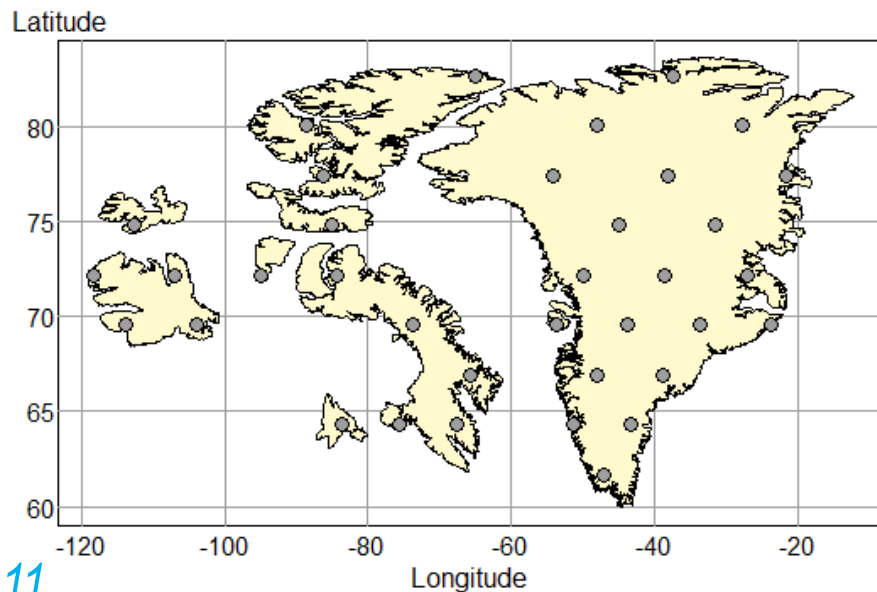
You can tell your friends that you are a data science analytics programmer.

Day LST Increase in Mainland SE Asia: Mar 2000 - Feb 2021
Latitude Degrees

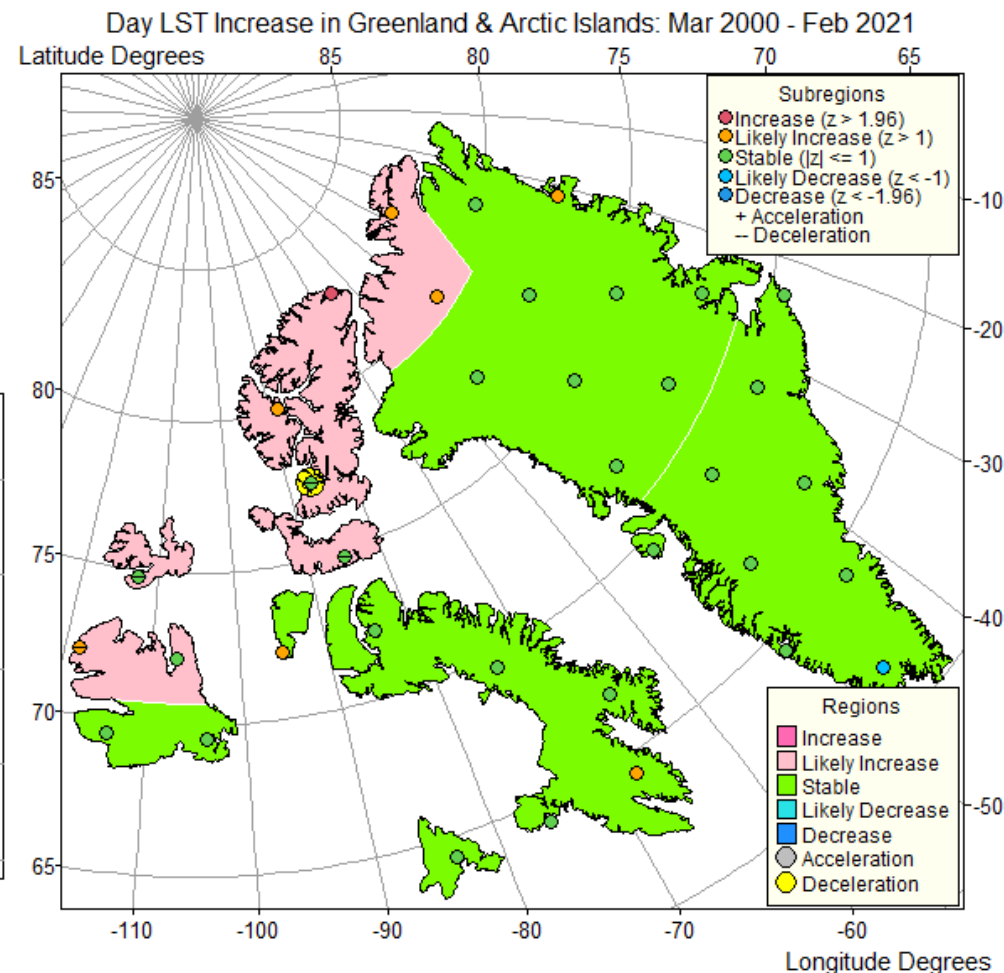


Your next task is to create a similar map for Greenland & sampled western islands. Do this by copying **greenland.Rcm** and **giLSTinc.txt**. and executing the **Rcm** file.

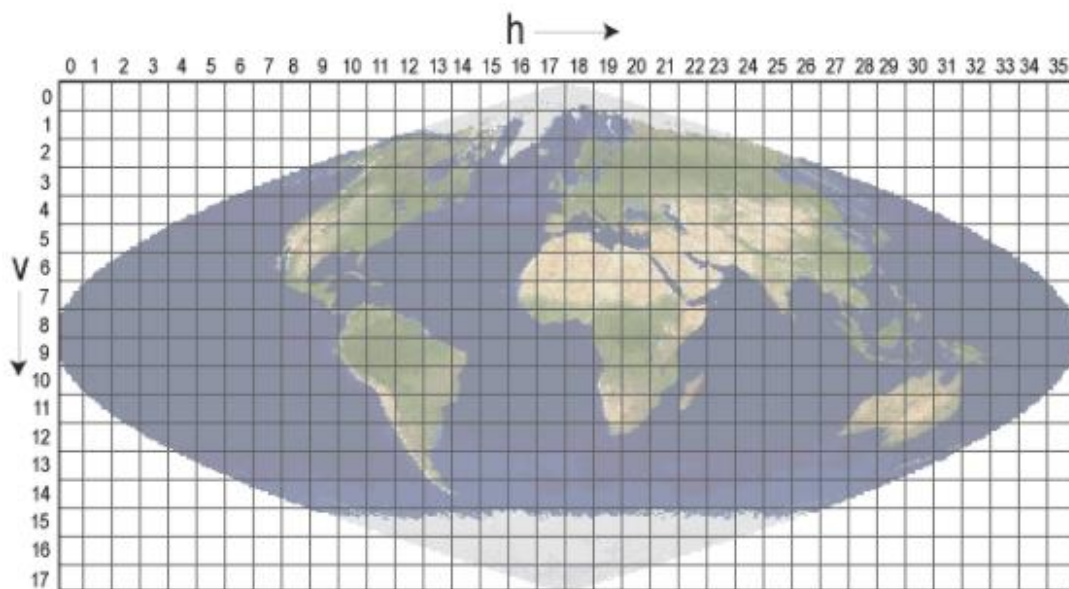
But this map isn't what the *Terra* satellite sees, because longitudes near the poles are much closer together than those near the Equator.



The map below is what Terra sees. Grid points we've been using for our survey of day LST increases are shown, and are equispaced, unlike the same points in the map on the left.



MODIS climate data are stored as **pixels** on a spherical surface approximating that of the Earth. For land surface temperature, these pixels all have the same west-east and north-south span (925.7 meters) and are referred to as 1 square kilometer pixels even though their area is only 0.857 km². They are grouped into **tiles** comprising arrays of 1200 x 1200 pixels into a **sinusoidal tile grid**. To identify a specific pixel, four coordinates are used. The first two (**v** and **h** in



the map on the left) denote the vertical and horizontal locations of the tile, and the second two (**line** and **samp**) denote vertical and horizontal pixel locations.

These coordinates can be converted to and from latitudes and longitudes using the URL

https://modis-land.gsfc.nasa.gov/MODLAND_grid.html

landweb.modaps.eosdis.nasa.gov/cgi-bin/developer/tilemap.cgi

To create a map similar to what the *Terra* satellite sees, we use what we call a “sinusoidal polar” projection. This is essentially the same as the sinusoidal tile grid described on Slide 12, but with its origin centered at a user-specified location that could be anywhere on the Earth’s surface. For example, the map shown on the right of Slide 11 has its origin at latitude 45 degrees north and longitude 80 degrees west of Greenwich. This feature allows us to create a map that looks directly down from outer space to the specified location.

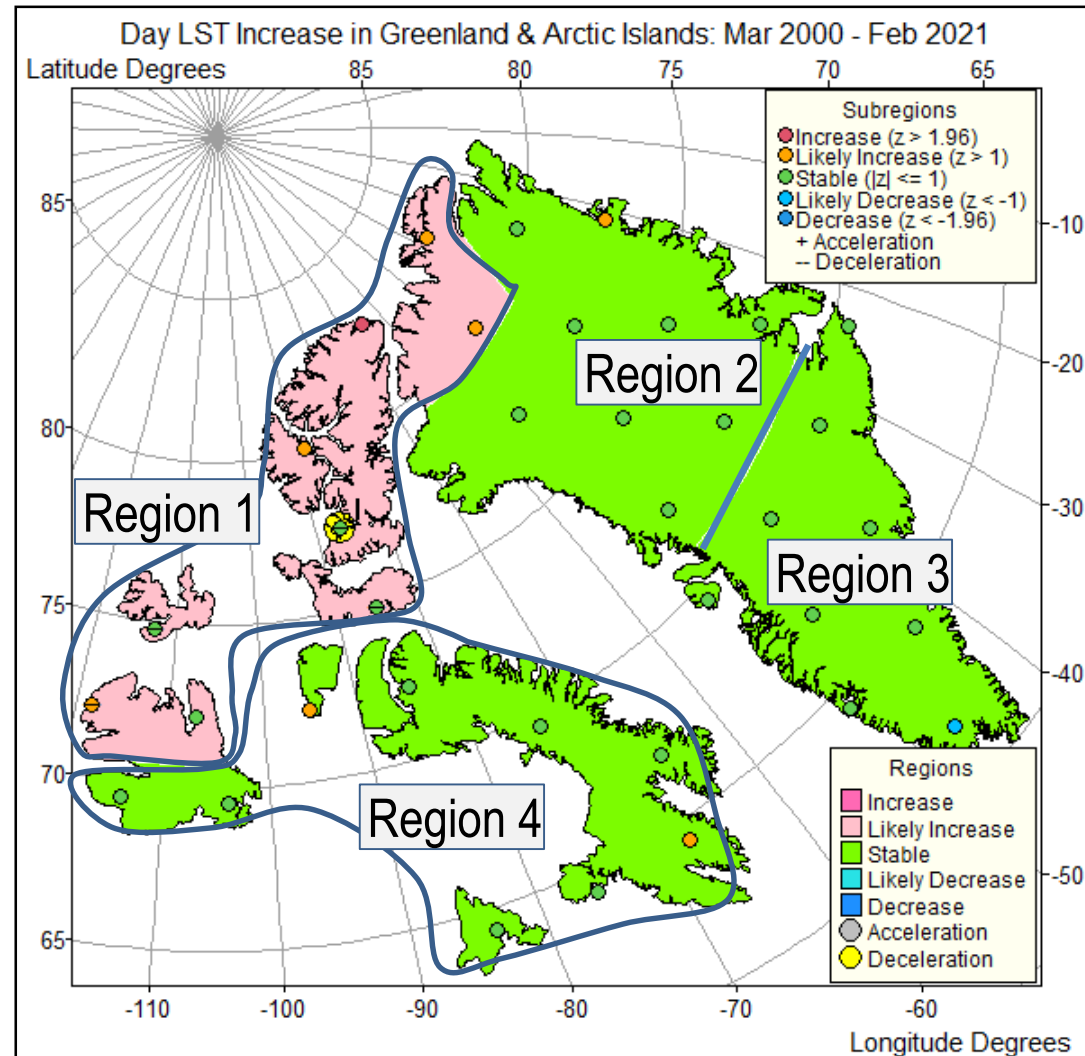
Here’s what the R code looks like. It just involves trigonometric transformations of the latitudes and longitudes.

```
phd0 <- 105; thd0 <- 50
wi02 <- subset ( wc, plotID==86)                # Greenland
xx <- wi02$x; yy <- wi02$y
ph <- (xx+phd0)*pi/180; th <- yy*pi/180
xC <- cos(th)*cos(ph); yC <- cos(th)*sin(ph); zC <- sin(th)
xCR <- cos(th0)*xC + sin(th0)* zC; yCR <- yC; zCR <- -sin(th0)*xC + cos(th0)*zC
lonR <- 90-(180/pi)*atan2( xCR, yCR); latR <- (180/pi)* asin (zCR)
lonR <- lonR*cos( latR*pi/180)
polygon( lonR, latR, col=rclr, border=gClr )
```

Your next task is to use the program [greenlandPolarA.Rcm](#) to create this corresponding map for Greenland and the western islands sampled in the grid. You'll also need to get a copy of the text file [wgis.csv](#) that contains boundaries for Greenland and these western islands that are in the global grid sample.

We've created four regions from the 36 grid points that fall upon land in Greenland and the north Canadian islands. These islands comprise Ellesmere, Devon, Melville and North Victoria in Region 1 and Baffin, Somerset, Southampton and South Victoria in Region 4.

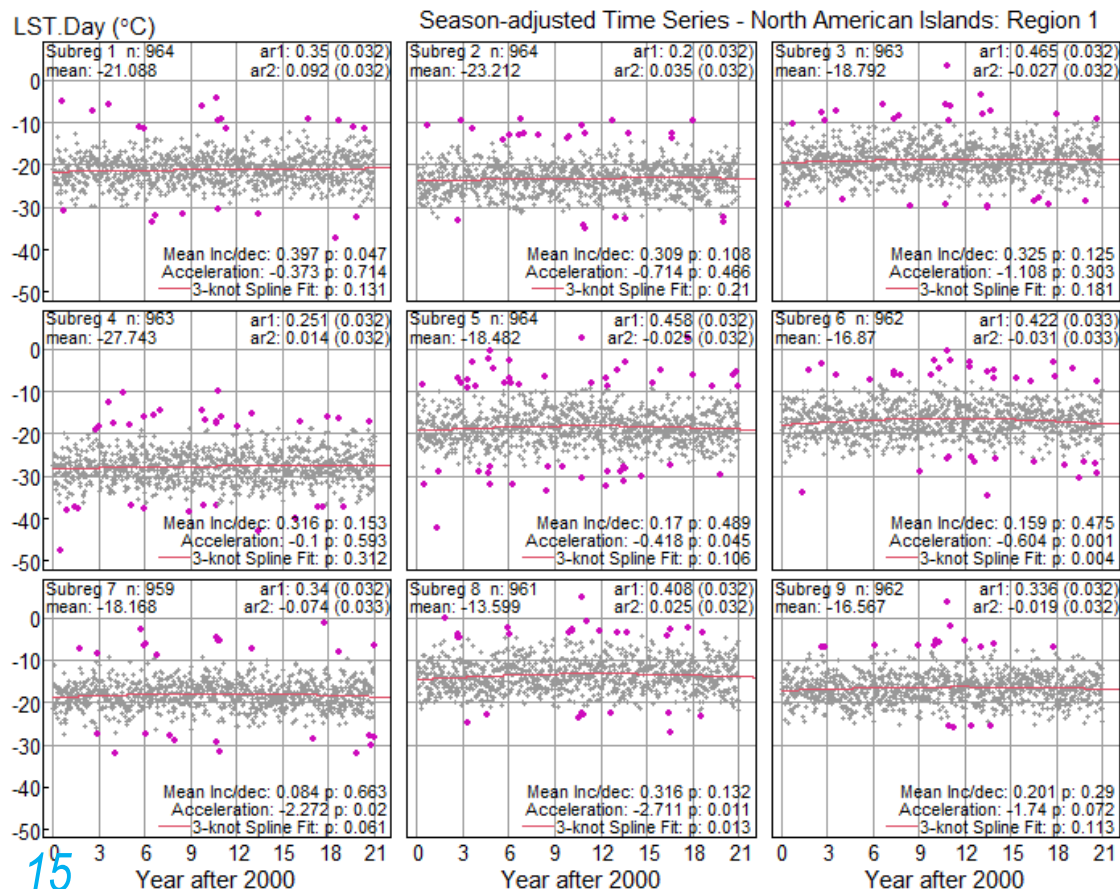
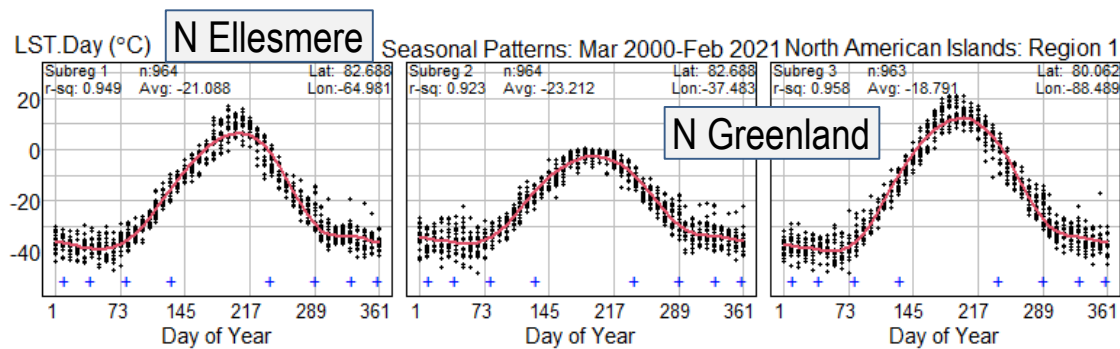
To include Canada, you'll also need the file [wca.csv](#).



NASA Data

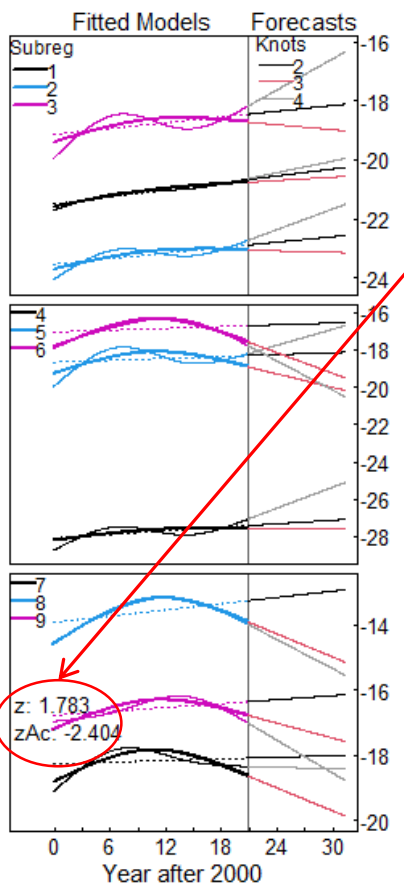
Region 1

Graphs of Models with Forecasts



Seasonal patterns for day LST in north Greenland & Ellesmere Island are much more variable than those in tropical SE Asia.

Time series for aggregated sub-



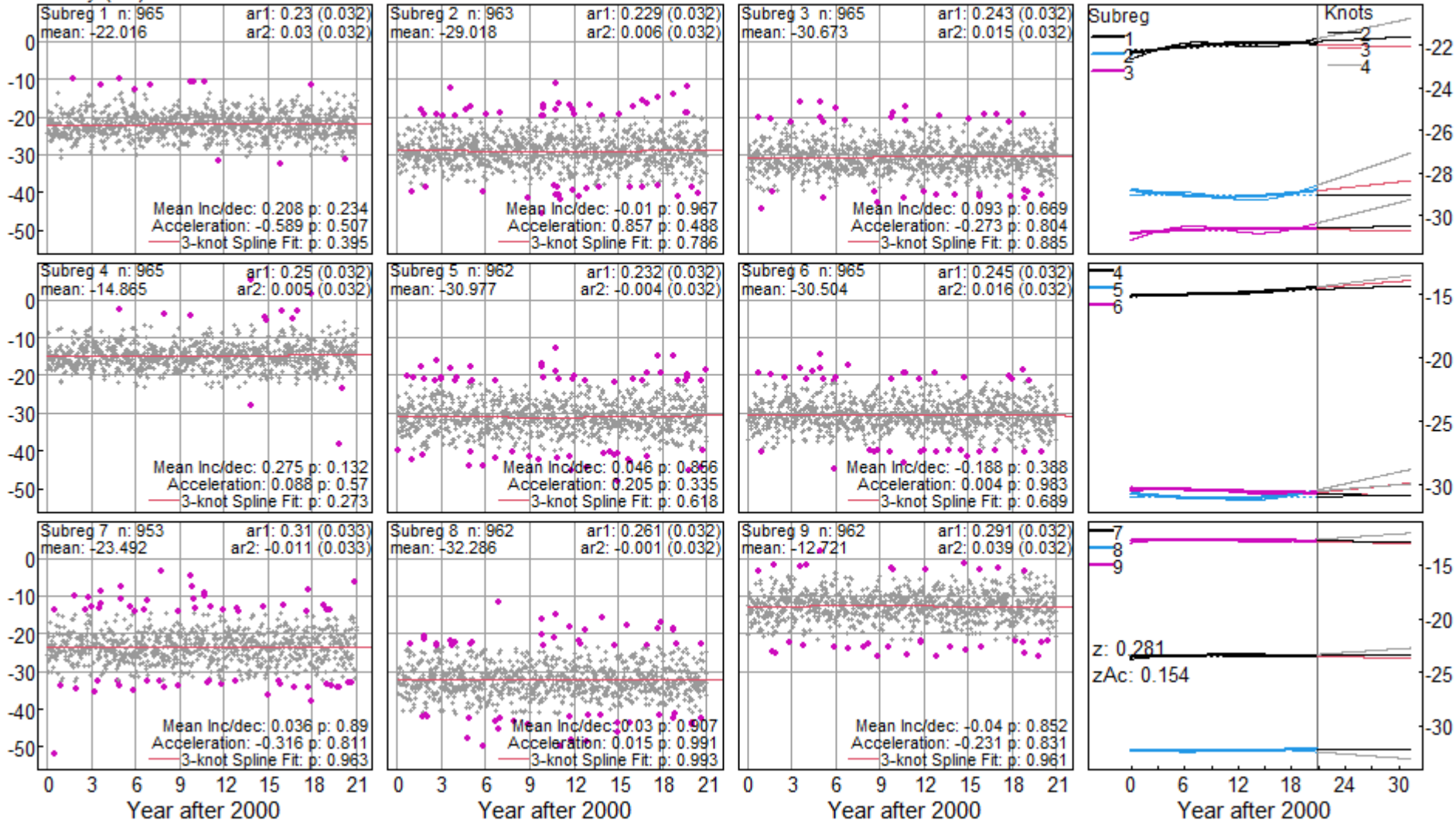
regions in this region show a likely increase, with $z = 1.78$, but day LST **decelerated** ($zAc = -2.4$), and forecasts suggest that this pattern will continue for several years.

Region 2: Central Greenland

LST.Day (°C)

Season-adjusted Time Series - North American Islands: Region 2

Fitted Models Forecasts



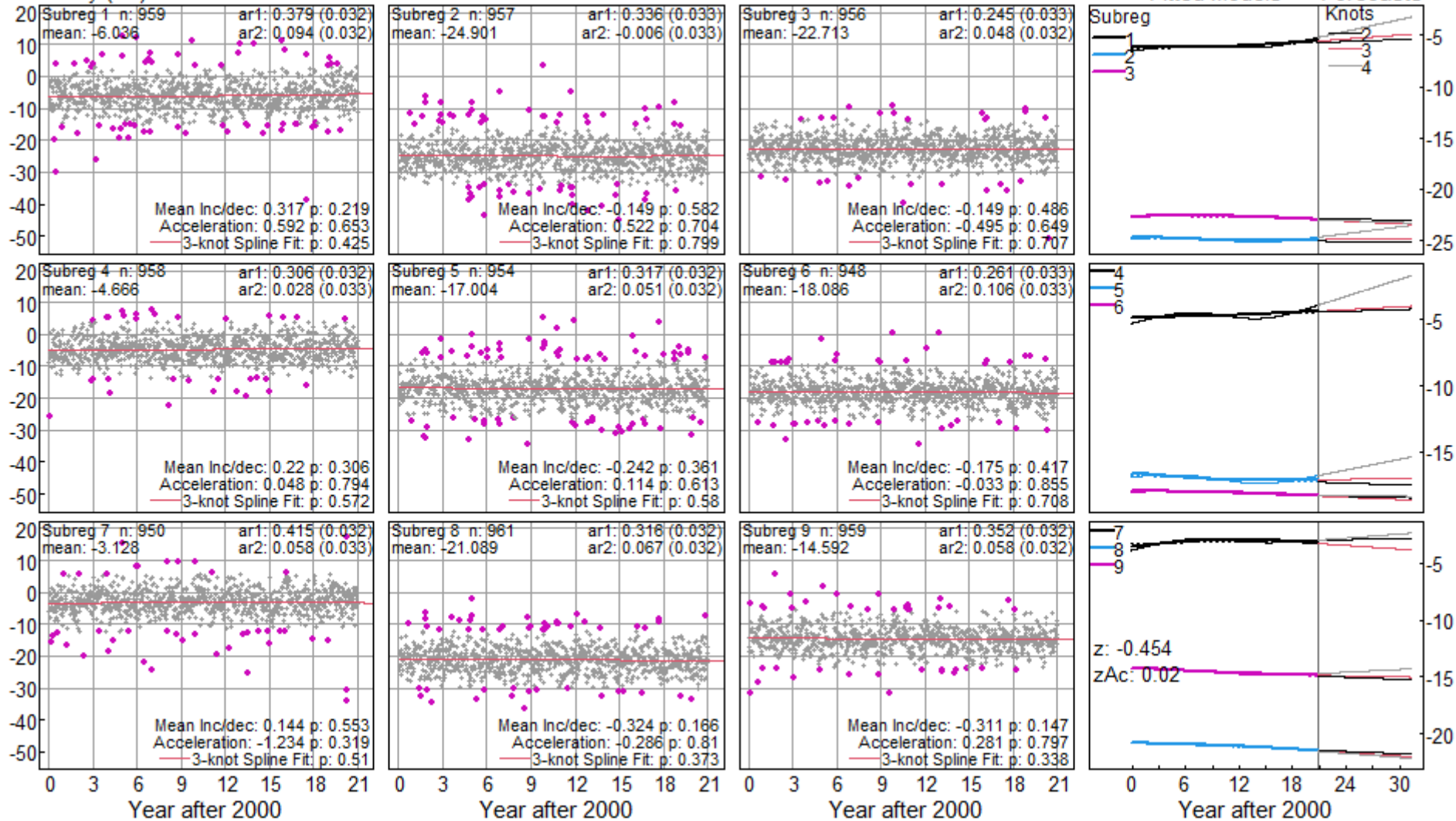
Climate scientists are concerned because global sea level will rise by 7.2 meters if its glaciers melt. But these plots show that day LST in central Greenland has been stable over the last 21 years, and is forecast to remain so.

Region 3: Southern Greenland

LST.Day (°C)

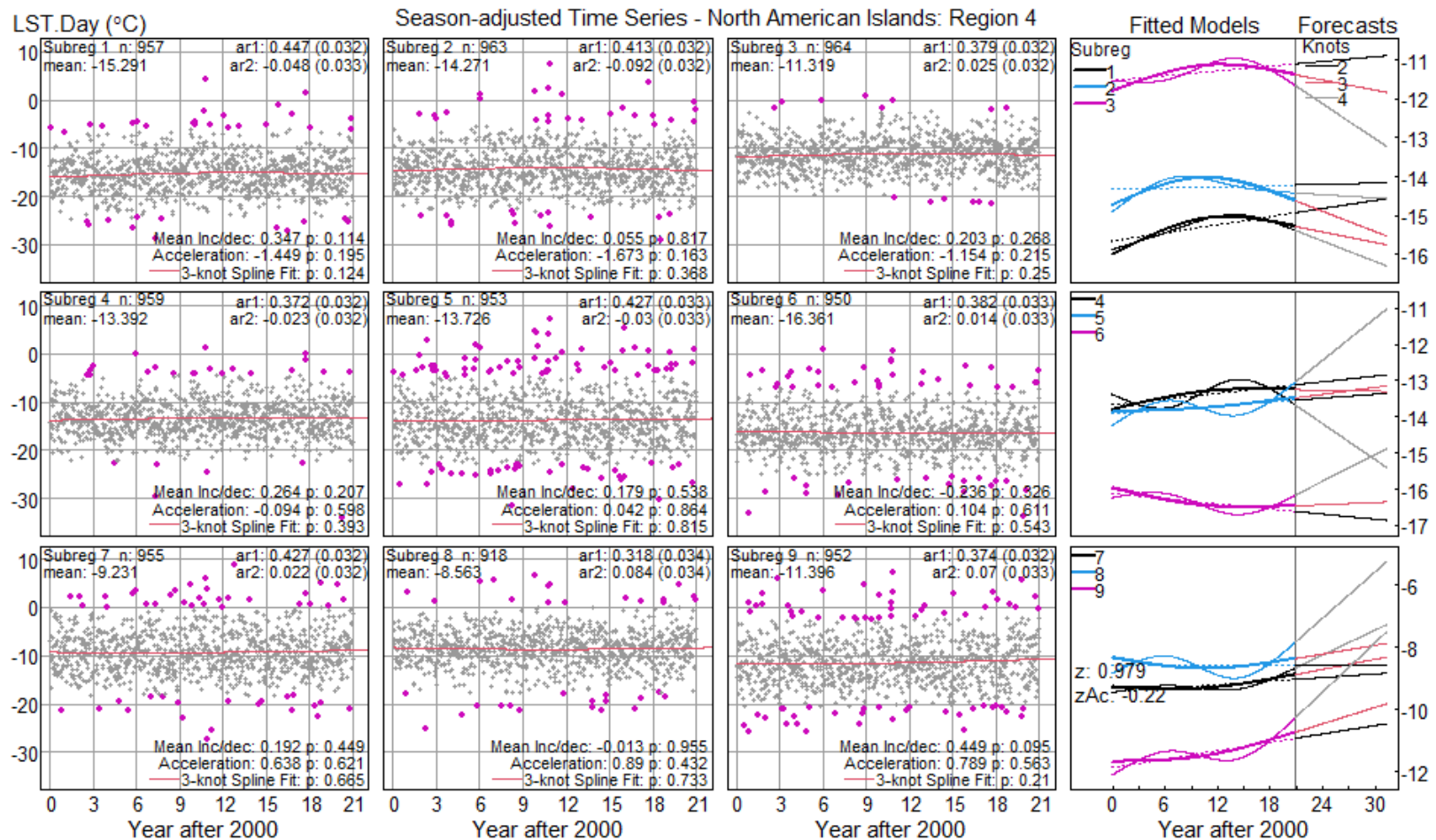
Season-adjusted Time Series - North American Islands: Region 3

Fitted Models Forecasts



These results complement those for the northern areas of Greenland, again showing stable day land surface temperatures with similar forecasts.

Region 4: Baffin, South Victoria, Somerset & Southampton Islands



In other major islands north of Canada day land surface temperatures did not appear to increase over the last 21 years. But lack of homogeneity suggests that this sample is too small to provide conclusive results, and that further analysis similar to what we did for the South-East Asian region is needed.

To summarize, we have continued applying basic data analytic methods to samples of daytime land surface temperature remote sensing data reported from Earth-orbiting satellites from March 2000 to February 2021.

In particular, we found that it is important to ensure that a population sample is homogeneous. If not, its size needs to be increased to make sure that all different components of the population are covered.

We also learnt how to use a simple trigonometric transformation of latitude/longitude coordinates to create maps that do not distort the shape of the area on the Earth's surface as viewed from a satellite.

We also studied land surface temperature increases and forecasts in Greenland and north Canadian islands.

Please email me at don.mcneil@mq.edu.au if you'd like to work with us on this important research topic.

Thank you for your patience. Hope to see you next week!

Data Analytic and Empirical Forecasting Methods using Smart Linear Regression: Session 3

Don McNeil

Emeritus Professor, Macquarie University, Australia

Prince of Songkla University, Thailand, 6 February 2022

National Aeronautics & Space Administration (NASA) Data

Recap of Session 2

Blanket Coverage: Data from Tonga

Results for Larger Sample in Baffin Island Region

Using a Function to simplify a Computer Program

Forecasts based on Empirical Results

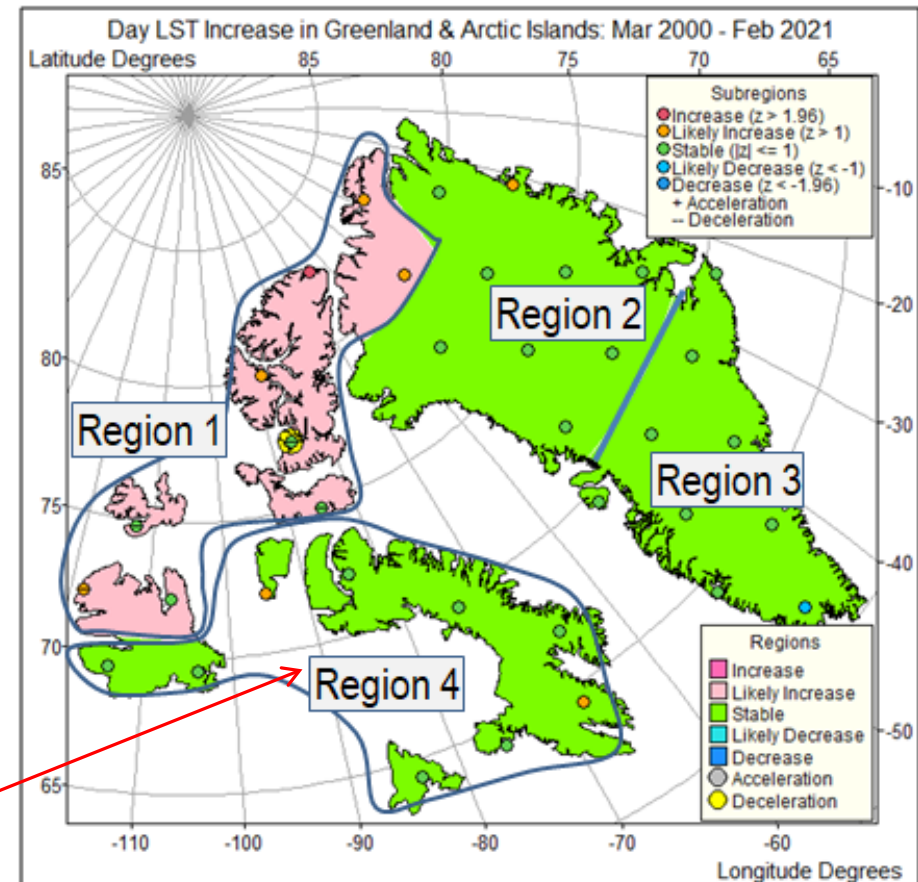
Take-home Message



Last week we continued to apply basic data analytic methods to land surface temperature (LST) data downloaded from a NASA website. We addressed the question “how big a sample is needed?”, showing that when the **homogeneity assumption** fails a much larger sample may be needed.

We showed how **forest plots** are informative when aggregating information from different samples taken from a population.

We used a **sinusoidal polar** projection to show what is really seen by a satellite above our planet. We studied daytime LST increase & acceleration in Greenland and major north Canadian islands. Is Region 4 homogeneous? We'll find out today.



NASA Data

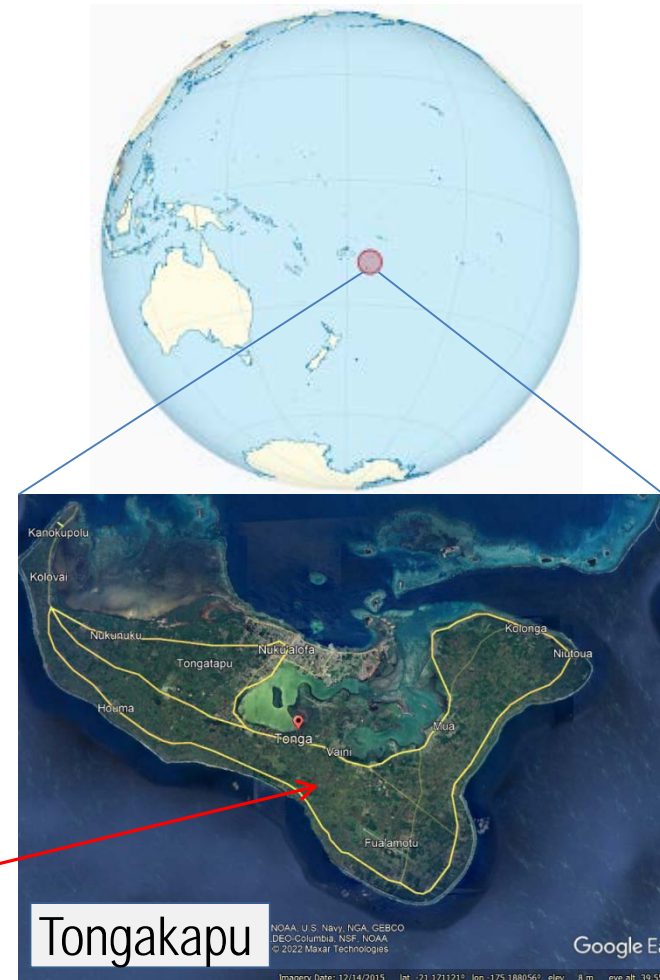
Blanket Coverage: Data from Tonga

Our climate research has focused largely on detailed blanket coverage of small areas, such as Nepal (Ira), Phuket province (Noppachai), south Asian islands (Tofan & Munawar) and Taiwan (Sahidan). But in an inspiring lecture at Hat Yai campus of PSU on 8 February 2018 Nobel chemistry laureate Fraser Stoddart convincingly advised us to "*tackle big problems*".

NASA climate data availability makes this possible, simply by taking a sufficiently large sample of the whole world. The regular grid we're using covers all land on the planet using a sample of no more than 2000 sub-regions each having area 42 km² and separated by a few hundred kilometers.

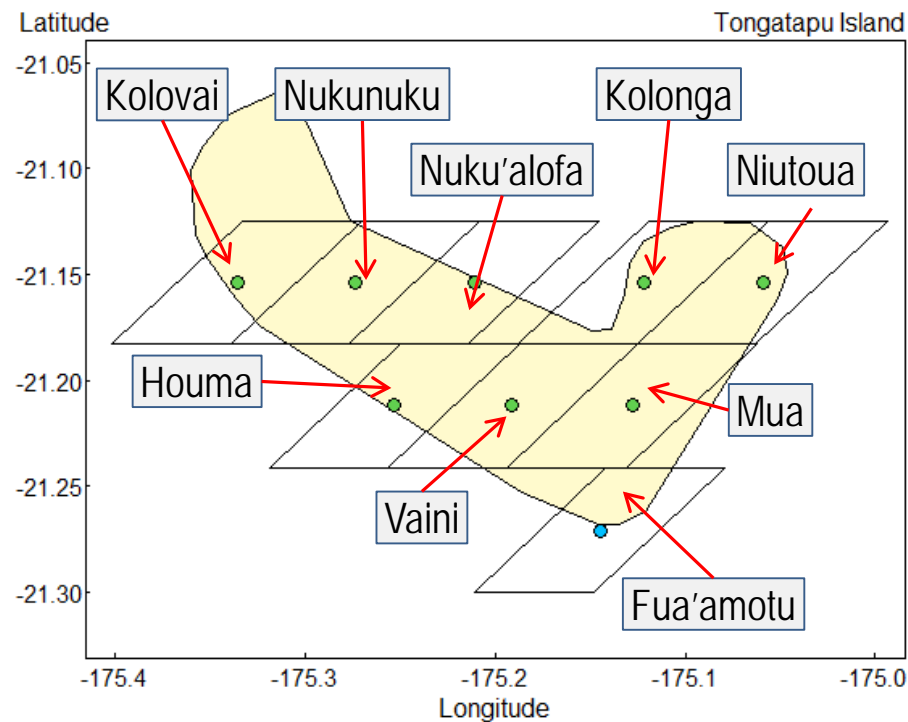
But if an area of interest is sufficiently small, we can still cover it with no gaps. **Tonga**, a group of islands in the Pacific ravaged recently by a tsunami caused by an undersea volcano, is such a place.

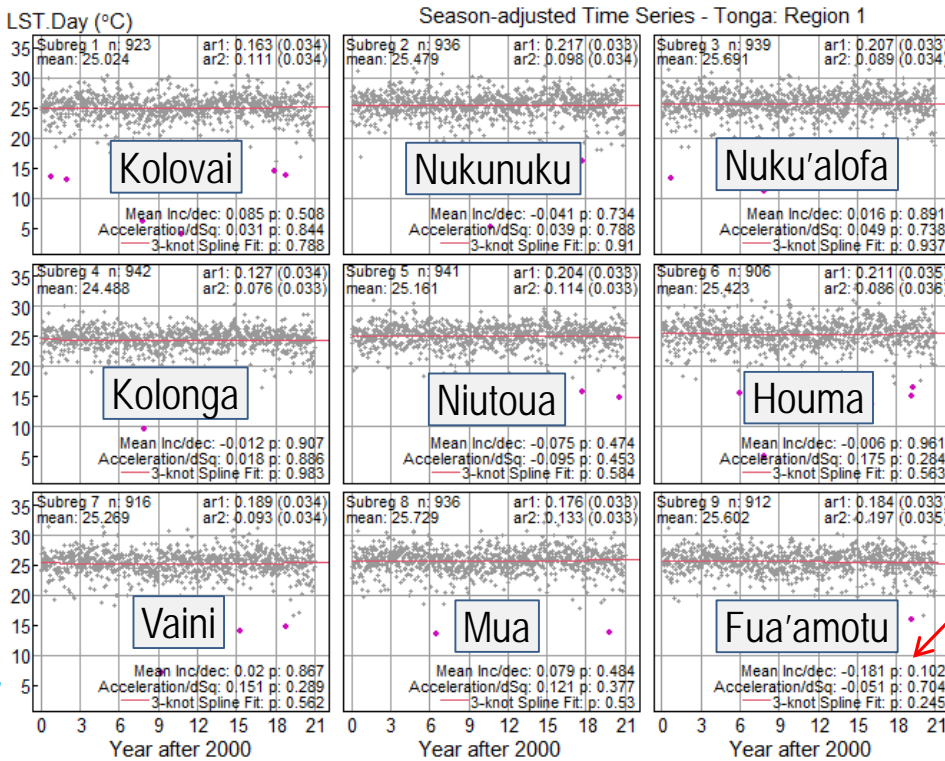
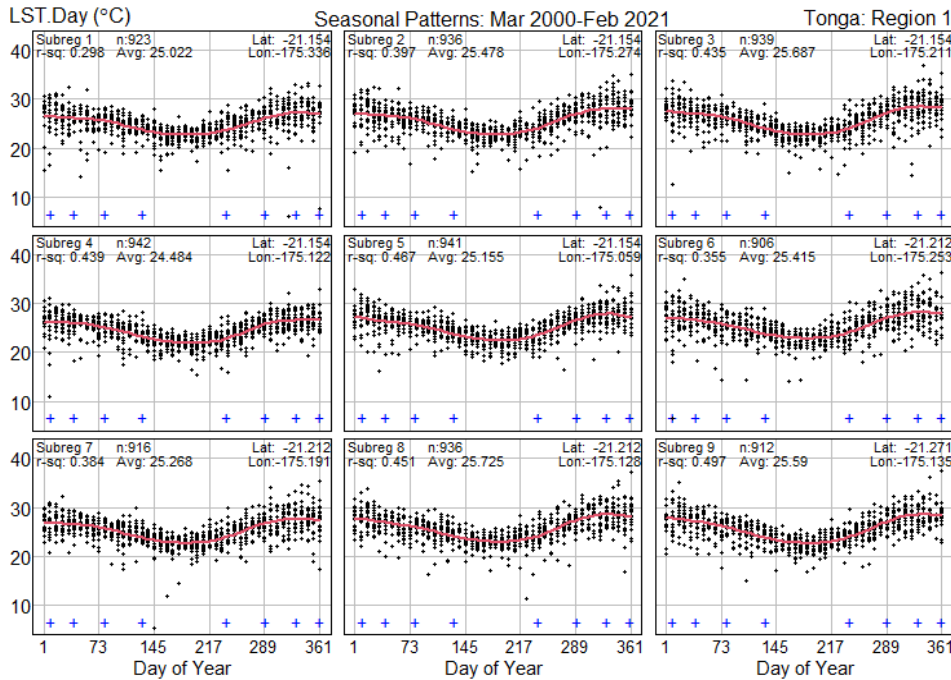
2 Its main island covers just 240 km².



Here's a map of nine sub-regions covering most of Tongakapu island. You can create this map by running the R program [tongaMap.Rcm](#) that reads files [wtg.csv](#) & [tg1LSTplnc.txt](#). The southernmost region covers a lot of water so most of its day LST data cannot be measured.

Note, however, that as long as at least one of the 49 pixels in each sub-region has a detectable value of day LST on any given day, this provides a measure of the average day LST for the whole sub-region on that day, although measurement accuracy is consequently reduced.





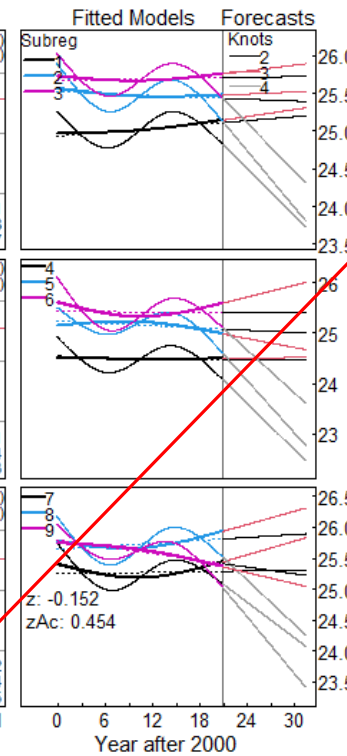
These graphs show results for the nine sub-regions of Tongakapu island mapped on Slide 3.

Given its tropical zone location, seasonal variation is not large.

Fitted models show very similar patterns for all nine locations, with no evidence of change except for

the southernmost sub-region, where a "likely" day LST decrease occurred.

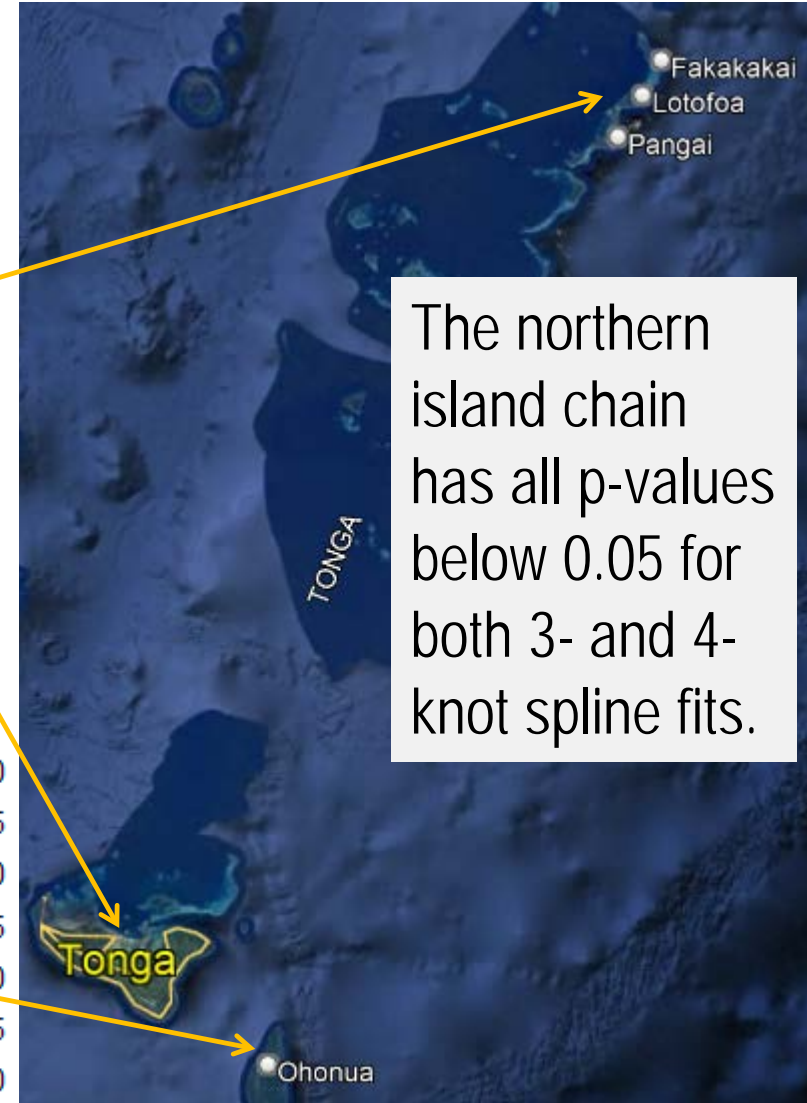
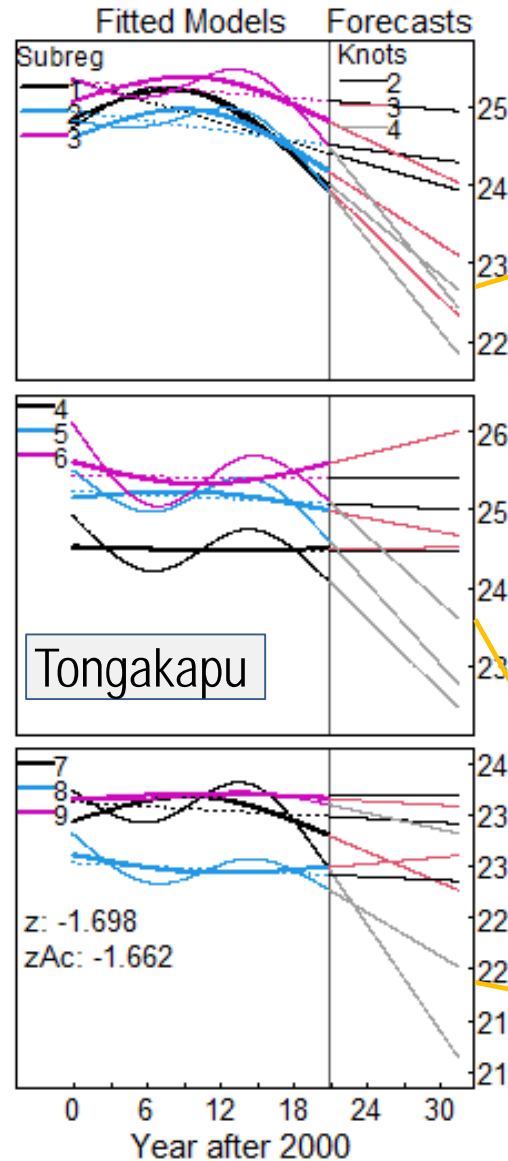
No 3-knot spline fit has a p-value below 0.05, but for the 4-knot spline fit sub-regions 4, 5, 6, 8 & 9 are all statistically significant.



These graphs show results from four more islands in Tonga (Ha'ano, Fortua and Lifuka in the northern group and Ohonua south of Tongakapu), together with those for Tongakapu already shown in the central panel of Slide 4.

These graphs show deceleration in day LST change for the three islands in the northern group.

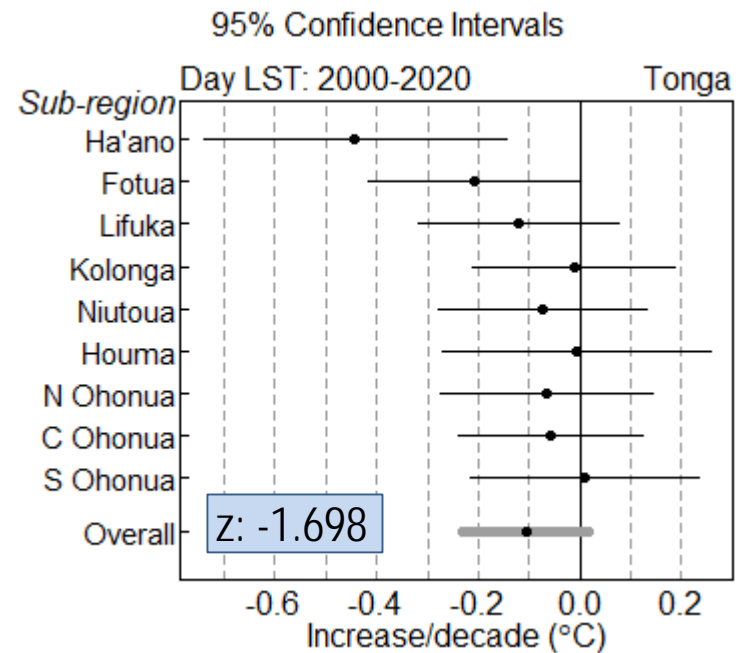
The 4-knot fits are quite similar, with most showing a 20-year cycle, but more extensive data are needed to confirm this pattern.



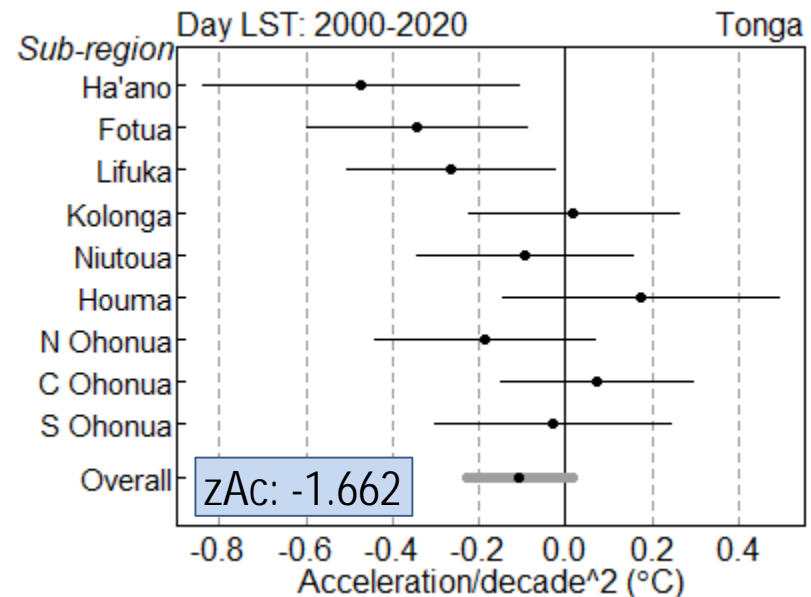
The northern island chain has all p-values below 0.05 for both 3- and 4-knot spline fits.

A forest plots of day LST increases in these island sub-regions of Tonga appears to satisfy the homogeneity assumption, with the possible exception of Ha'ano in the northern chain. Results suggest "likely decrease" ($z = -1.7$).

Note that acceleration, like what a car does when the driver puts the foot down, is the gain in increase per unit time, in this case degrees Celsius increase *per decade per decade* ($/\text{decade}^2$). This can be estimated from the data using the 3-knot spline, defined in Slide 14 of Zoom 1 as $y = a + bx + cs_1(x)$. The spline $s_1(x)$ reduces to the linear function $d + 3cx$ beyond the third knot, so the acceleration per decade^2 is obtained by multiplying the estimate of c by 3 and dividing by 2.1 (decades observed).

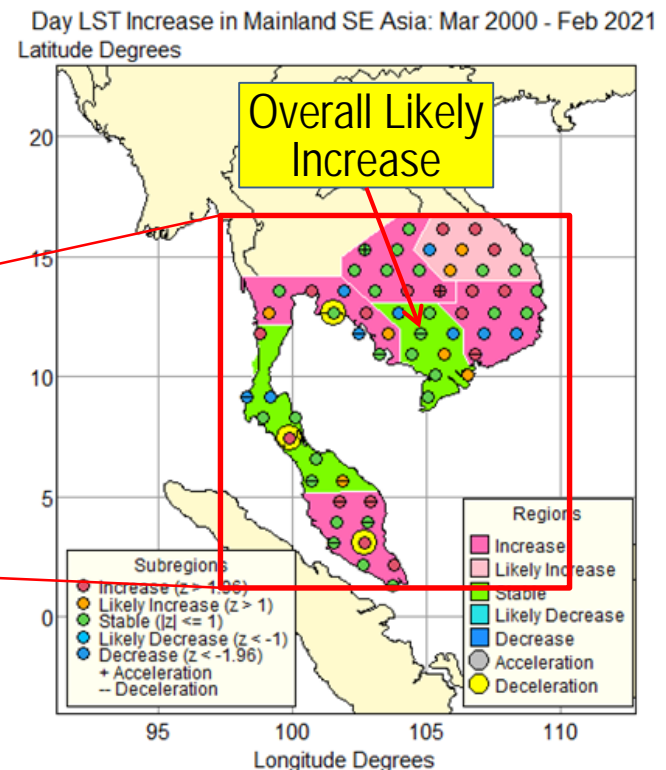
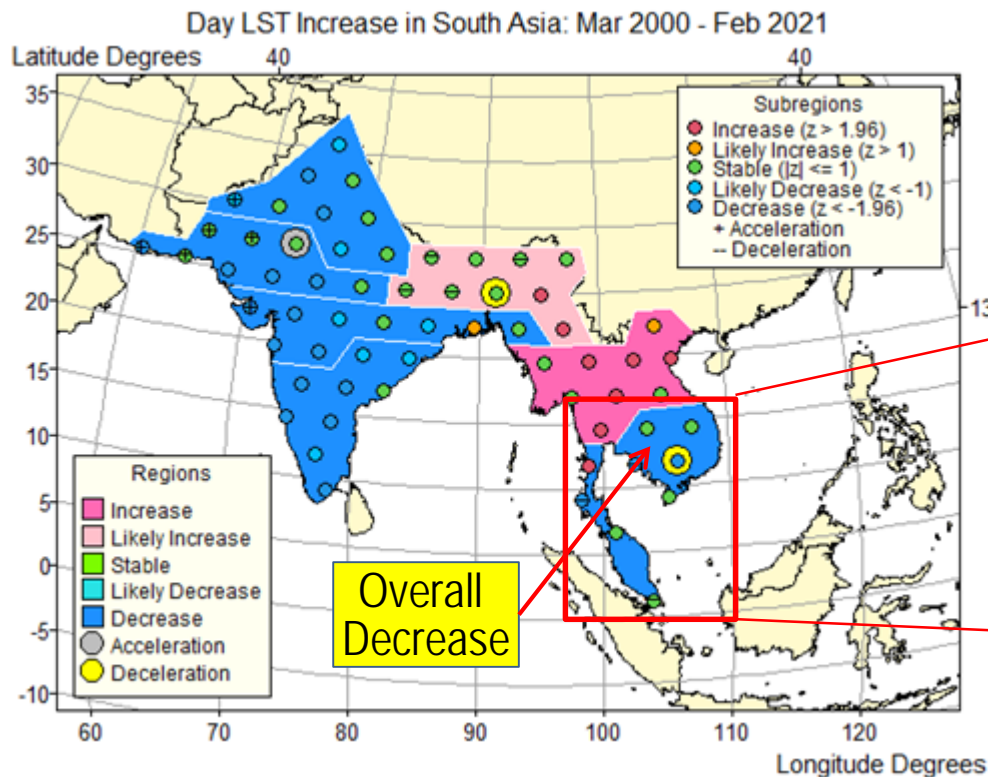


Get [tg2LSTplnc.txt](#) and run [tongaCplots.Rcm](#) to create these.



Last week we got more accurate results for an inhomogeneous region in South-East Asia by splitting it into seven smaller regions each containing nine sub-regions. As a result, we concluded that day LST in the region overall was

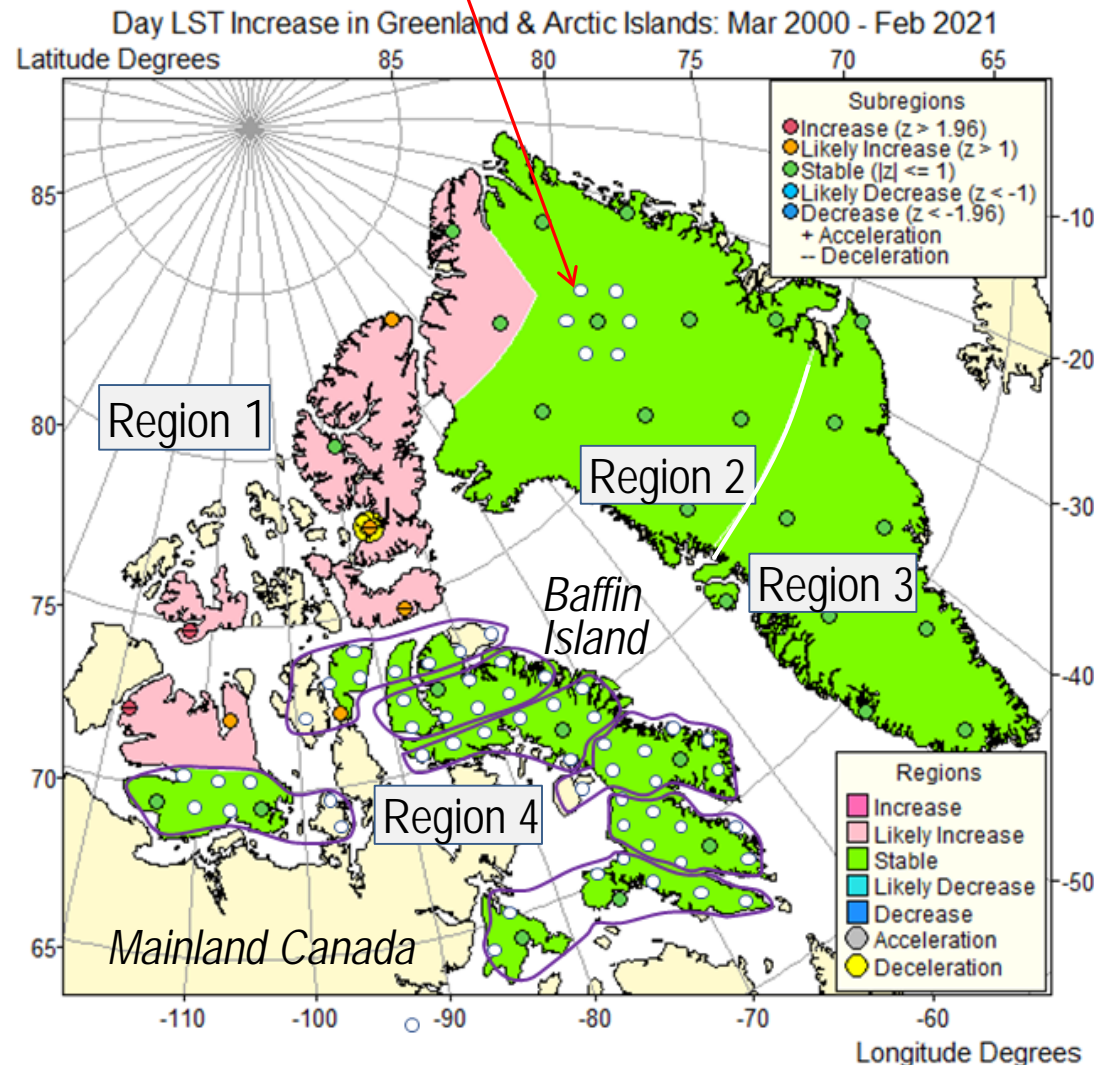
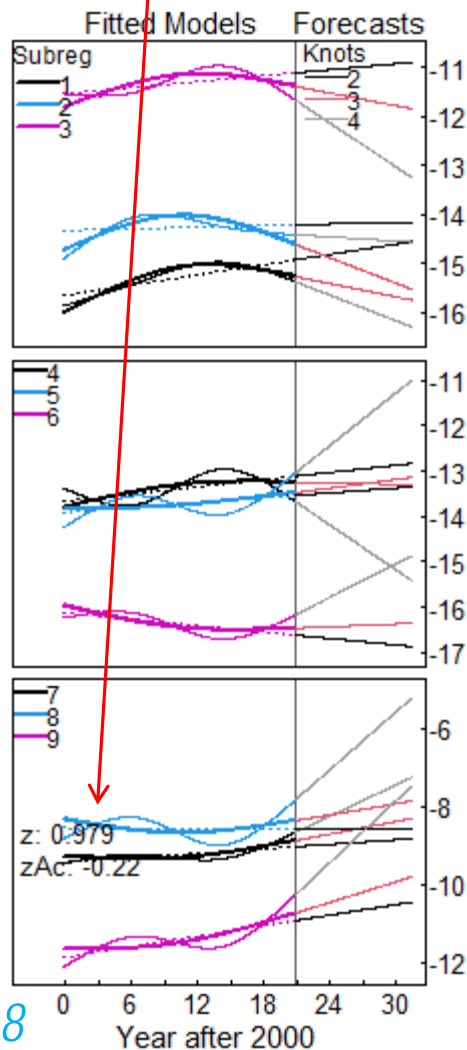
“likely increase”, not “decrease”.



We'll now use the same method to re-examine day LST change in region
7 comprising Baffin Island and islands to its west in northern Canada.

The graph below shows that in Region 4 (comprising Baffin, south Victoria and some other islands in northern Canada) day LST was stable from 2000-2020, with z-value 0.979.

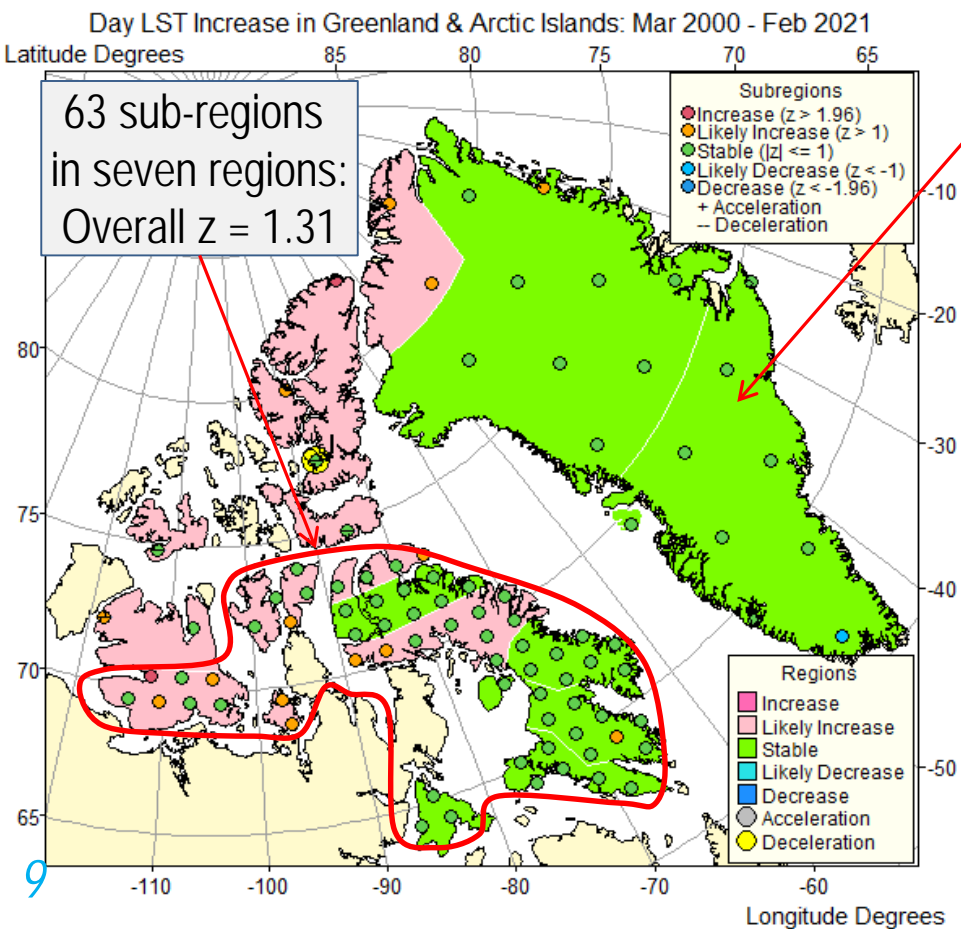
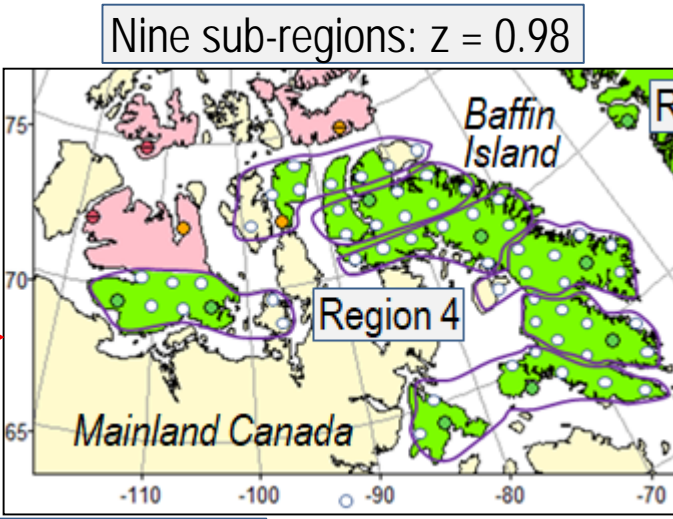
Note that we can insert two extra sub-regions between each neighbouring pair.



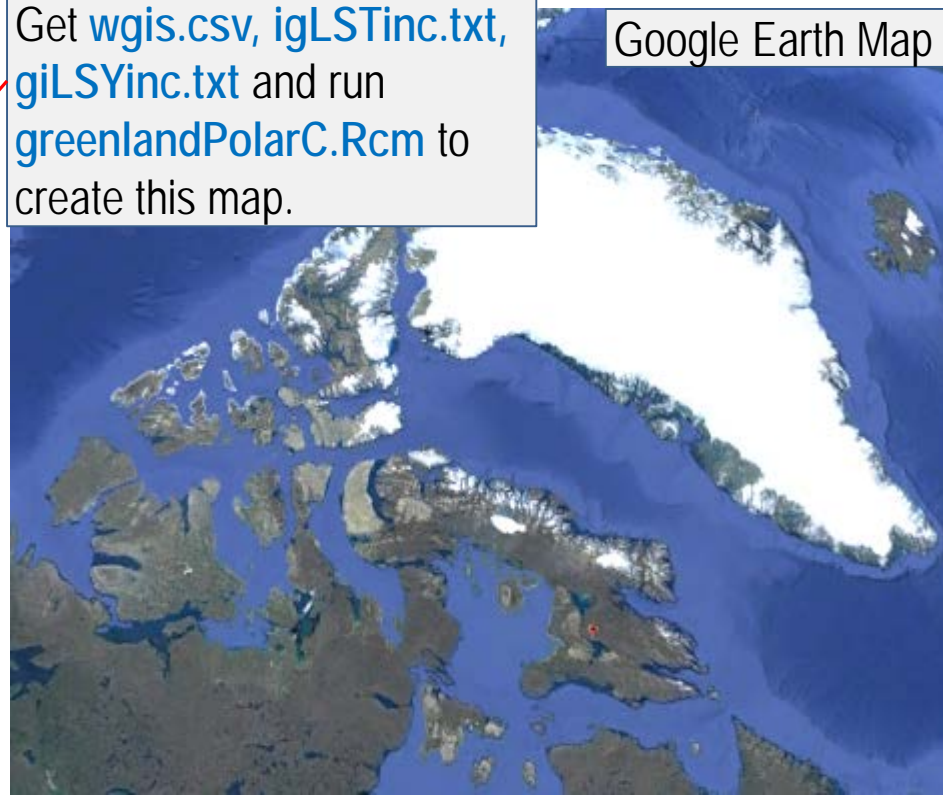
This is expected to increase the sample by a factor of seven, giving rise to 7 smaller regions and covering additional islands.

A schematic map shows “likely increase” in three of the seven smaller regions and “stable” day LST in the other four regions, corresponding to an overall z-value of 1.31 (“likely increase”)

This compares with $z = 0.98$ (“stable” LST) in the original analysis.

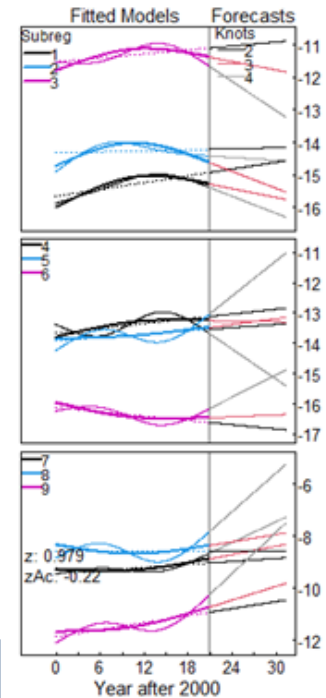
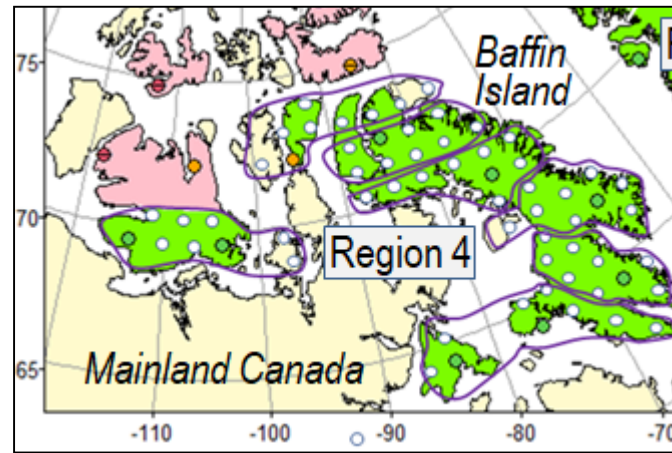


Get [wgis.csv](#), [igLSTinc.txt](#), [giLSYinc.txt](#) and run [greenlandPolarC.Rcm](#) to create this map.



Plots of fitted models show wide variation for day LST increase and acceleration in different regions.

Nine sub-regions: $z = 0.98$



63 sub-regions in seven regions: Overall $z = 1.31$

North-West

North-East

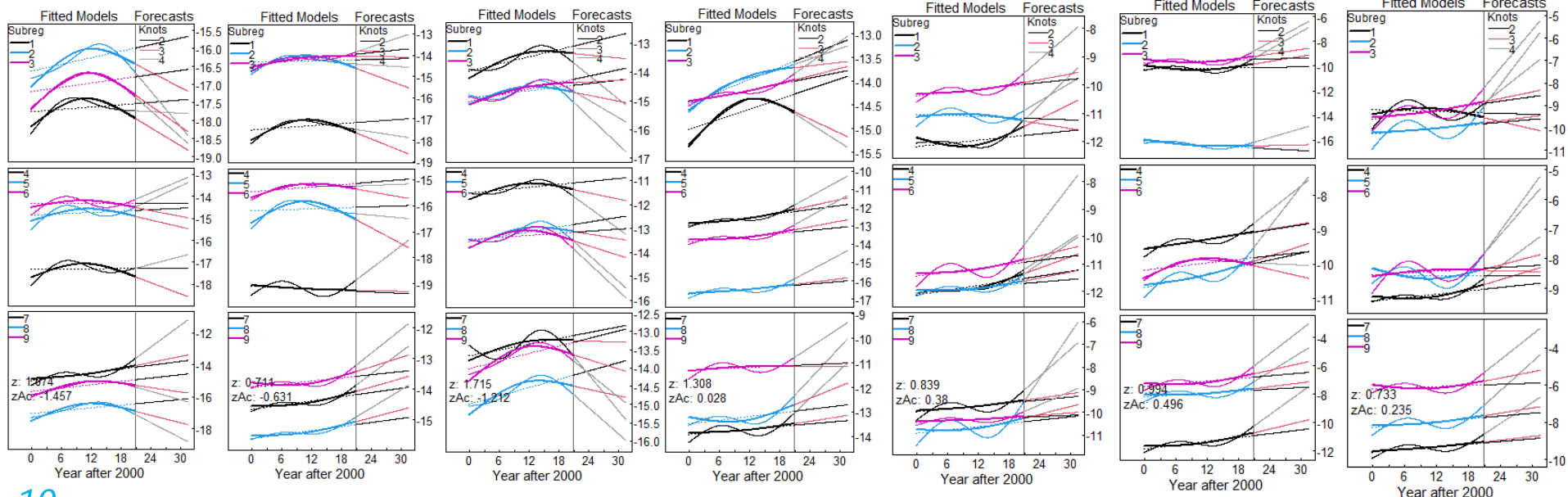
Far West

Central

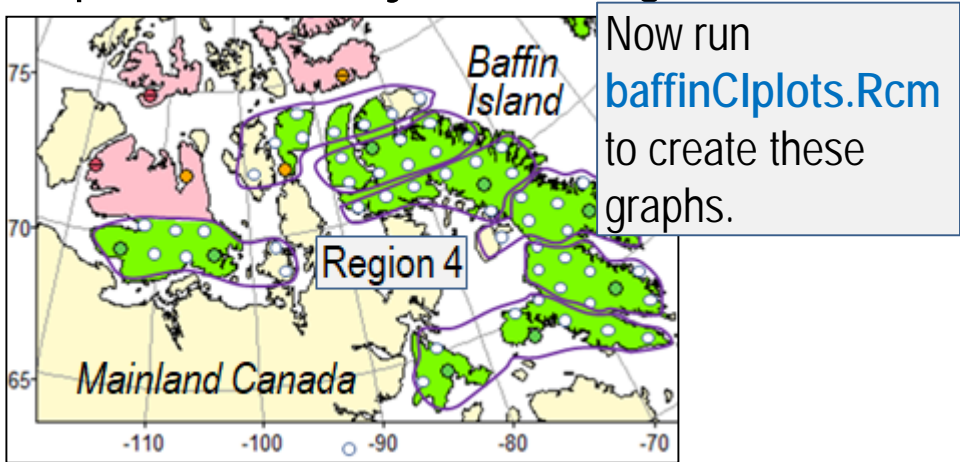
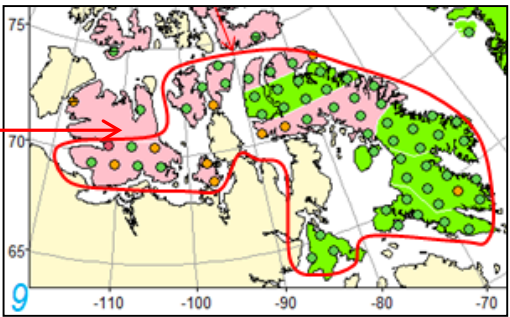
East

Upper South

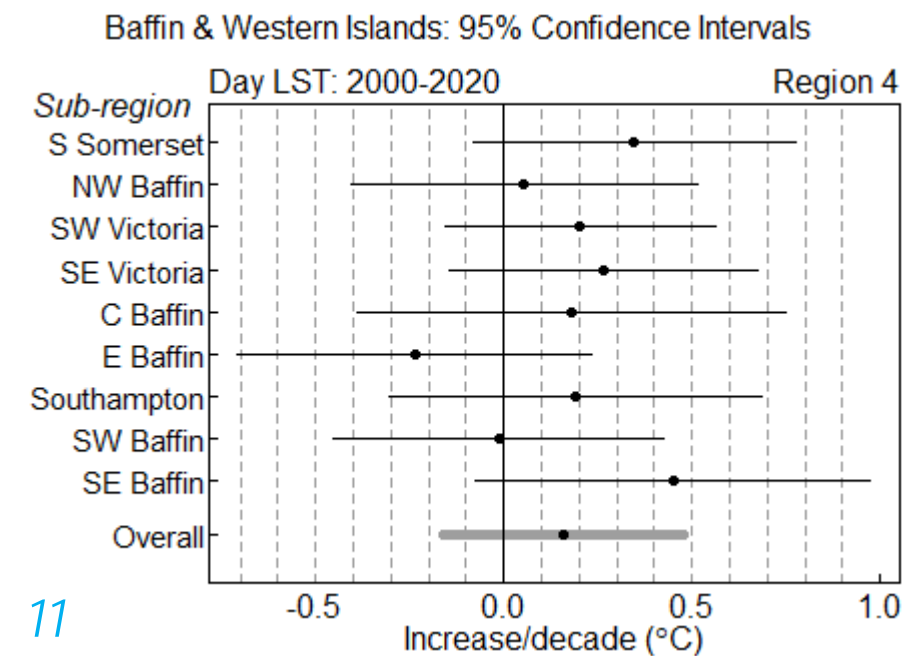
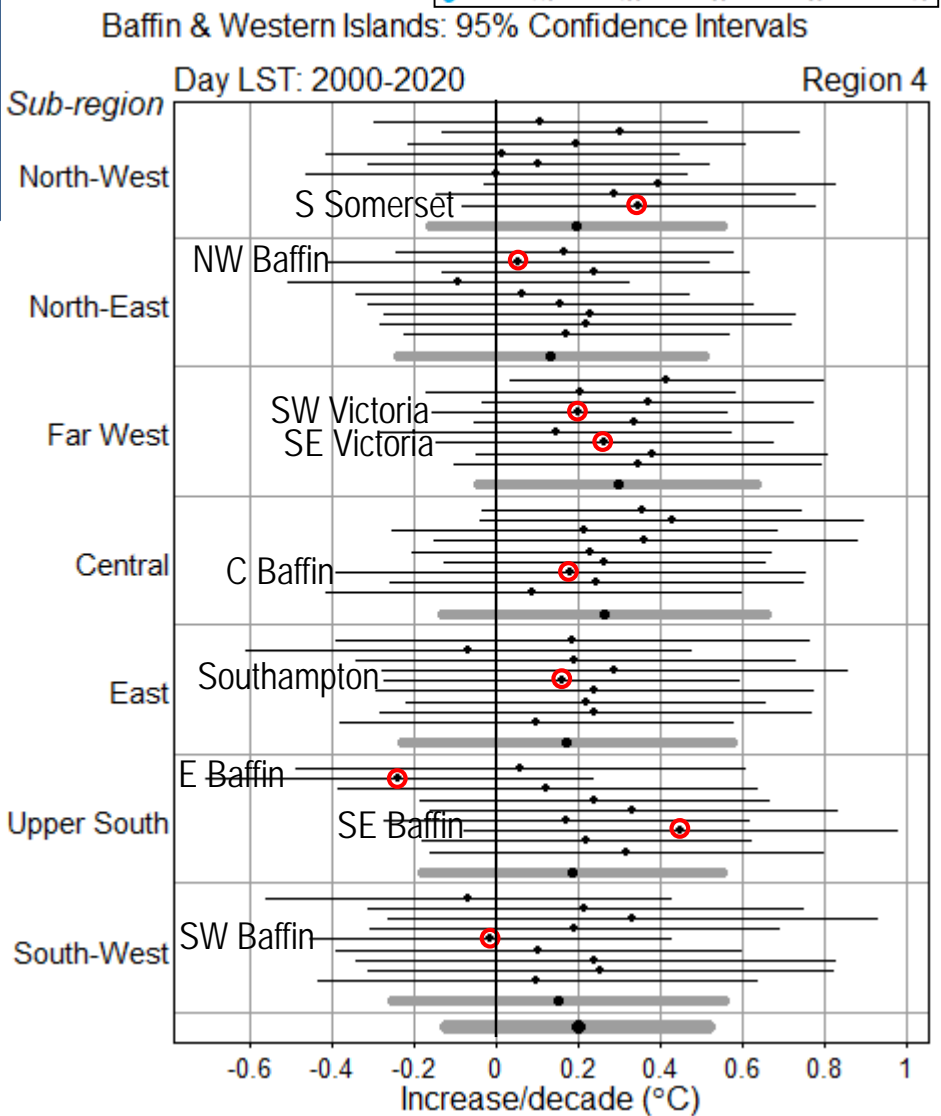
South-West



Confidence intervals also highlight wide variation but no indication of inhomogeneity. We see that Victoria Island need not have been divided. The next step would be to repeat this analysis for Region 1, including more islands.



Now run `baffinCIplots.Rcm` to create these graphs.



The code to create a map of what the *Terra* satellite sees is repeated many times in the program [greenlandPolarA.Rcm](#) that we used to display the map on Slide 9. It contains the following R commands, where [wa](#) is a data frame containing the four column variables [plotID](#), [pointID](#), [x](#) and [y](#) to be mapped.

```
wa$x <- wa$x+phd0      # longitudes with origin at phd0
ph <- wa$x*pi/180      # converted to radians
th <- wa$y*pi/180      # latitudes in radians
xC <- cos(th)*cos(ph)  # their Cartesian x coordinates
yC <- cos(th)*sin(ph)  # their Cartesian y coordinates
zC <- sin(th)          # their Cartesian z coordinates
```

```
xCR <- cos(th0)*xC + sin(th0)*zC  # Step 1: rotate around axis through Equator
yCR <- yC
zCR <- -sin(th0)*xC + cos(th0)*zC
```

```
lon <- 90-(180/pi)*atan2(xCR,yCR)  # Step 2: convert back to longitudes and latitudes
lat <- (180/pi)*asin(zCR)
lon <- lon*cos(lat*pi/180)
```

We can create a function called `spp()`, say, that creates the sinusoidal polar longitude and latitude coordinates corresponding to `x` and `y` as follows.

```
spp <- function(x,y,phd0,thd0) {                                # sinusoidal polar projection function
  ph <- (x+phd0)*pi/180; th <- y*pi/180
  xC <- cos(th)*cos(ph); yC <- cos(th)*sin(ph); zC <- sin(th)
  xCR <- cos(th0)*xC + sin(th0)*zC; yCR <- yC; zCR <- -sin(th0)*xC + cos(th0)*zC
  lonR <- 90-(180/pi)*atan2(xCR,yCR); latR <- (180/pi)*asin(zCR)
  lonR <- lonR*cos(latR*pi/180)
  cbind(lonR,latR)
}
```

Here's how it works. Suppose `wgis.csv` is a CSV file in your working directory that contains boundaries for countries or islands you wish to map. Three of these places have plotIDs 35.166, 35.124 and 35.131. The origin for your map is at longitude 90 and latitude 50 degrees.

```
read.csv("wgis.csv", header=TRUE, as.is=TRUE) -> wc
phd0 <- 90; thd0 <- 50
wiz <- subset(wc, plotID %in% c(35.166,35.124,35.131))
lonlat <- spp(wiz$x,wiz$y,phd0,thd0)
polygon(lonlat[,1],lonlat[,2],border=1,col=rclr)
```

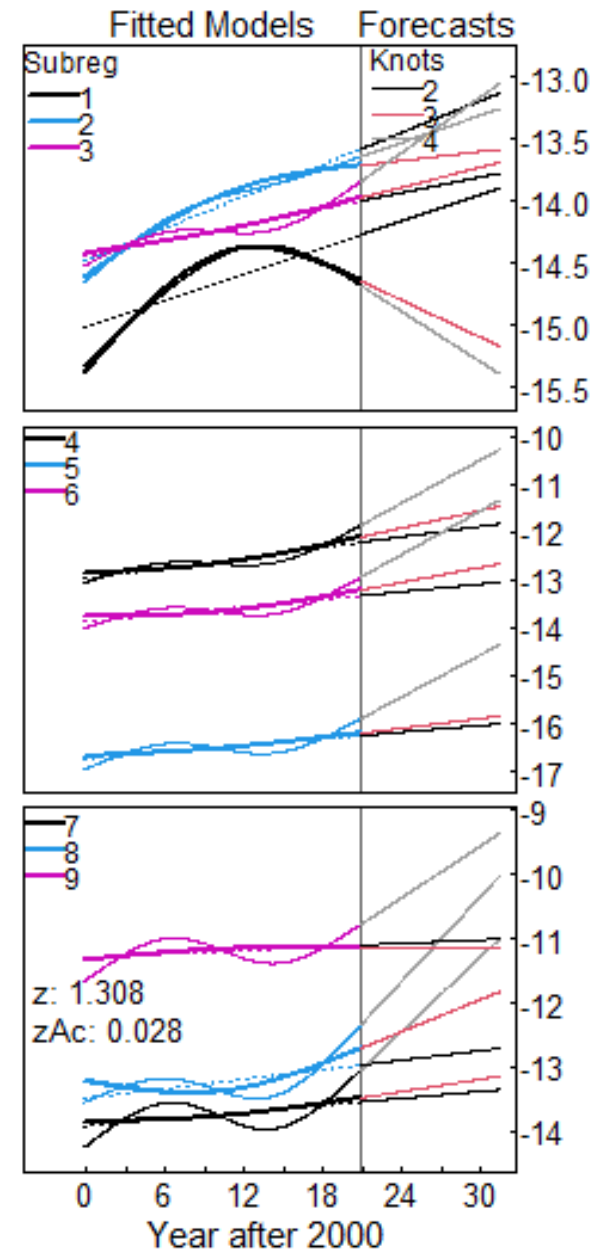
Climate trends are difficult to forecast. Nobody has yet come up with a theory that gives accurate results.

So instead, we'll just use the 21 years of NASA data to see if we can forecast a few years ahead, say 6 years.

The graphs on the right show fitted spline curves to season-adjusted day LST for the nine sub-regions of Central Baffin Island, with forecasts obtained simply by projecting the spline curves 10.5 years ahead.

Assume we have data for just 15 years and we fit a spline model to these data and use it to forecast the fitted value 6 years ahead, and compare the result with the known fitted value. Using a sizable sample of sub-regions, we can use the distribution of these errors to calculate a 95% confidence interval.

This is our **empirical** forecasting method.



However, this is easier said than done. The program we have been using assumes precisely 21 years of data, and would need to be generalized to allow different time spans. And it needs to be applied to sufficiently large samples of homogeneous sub-regions to achieve accurate forecasts. As we have seen different grid dimensions are needed in different areas of the world. And the amount of data is quite large.

Given that essentially the same program is used for all samples (`iaTDb5.Rcm` for the south Asian sub-continent, `giTDb5.Rcm` for north American islands, `tgTDb5.Rcm` for Tonga, etc.) it would be better to turn this program into a function, with arguments specifying parameters that specify the sample identity, the observation and forecast periods, and other relevant choices.

Once this is done, the function can be stored as an R command file (`tdb5.Rcm`, say) in a working directory and used in a another program simply by executing the statement `source("tdb5.Rcm")` within that program. If you have been using R programs to make graphs of democratic confidence you will be familiar with this method, where the function `dcis.Rcm` is used.

Using this approach, we'll show some empirical forecasts next week..

In this session we continued applying basic data analytic methods to samples of daytime land surface temperature remote sensing data reported from Earth-orbiting satellites from March 2000 to February 2021.

We saw that blanket coverage of all data in an area is feasible for small areas such as the Pacific islands of Tonga, where we found very similar trends in widely separated locations.

We also studied land surface temperature increases and forecasts in and around Baffin island, finding that results did not change very much when the sample size increased, in contrast to an area around southern Thailand.

And we saw how a user can create their own function to simplify computer programming, and we suggested how this approach could be used to facilitate empirical time series forecasting. Next week we'll follow up on this.

Please email me at don.mcneil@mq.edu.au if you'd like to work with us on this research topic.

Thank you for your patience. Hope to see you next week!

Data Analytic and Empirical Forecasting Methods using Smart Linear Regression: Session 4

Don McNeil

Emeritus Professor, Macquarie University, Australia

Prince of Songkla University, Thailand, 13 February 2022

National Aeronautics & Space Administration (NASA) Data

Recap of Session 3

Estimating Acceleration: The 3-Knot Spline

A Simulation Study

Function to Fit a Model to LST Data

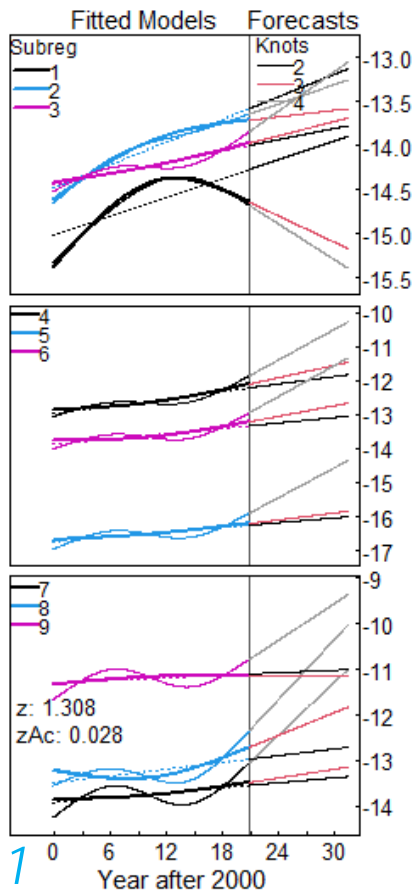
Empirical Forecasts

Take-home Message



NASA Data

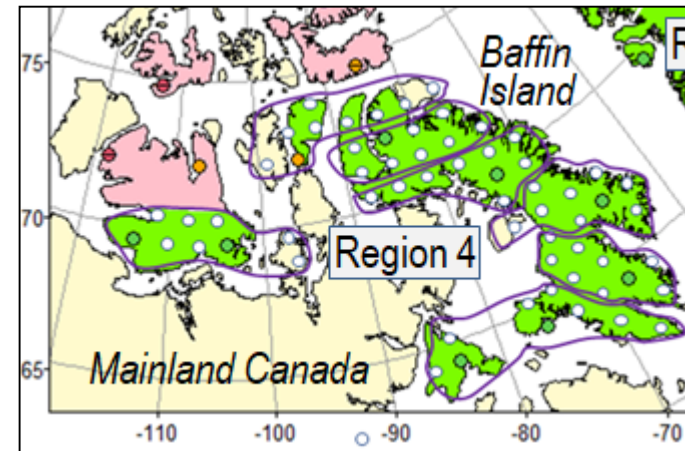
Last week we continued to apply data analytic methods to land surface temperature (LST) data downloaded from a NASA website, focusing on data from islands within the Arctic Circle above Canada and Tonga in the south Pacific ocean.



We also learnt how a **computer function** can simplify complex analysis, such as using natural cubic splines to analyse time series data.

In this session we'll show how such functions can simplify **empirical forecasting** of time series data.

Recap of Session 3



In Session 1 we gave a formula for a *natural spline*, defined as a piecewise cubic function that is linear beyond the range of the data. Two boundary conditions are needed to achieve this linearity, and the formula is as follows.

$$y = \mathbf{a} + \mathbf{b}x + \sum_{k=1}^{p-2} \mathbf{c}_k s_k(x),$$

where $s_k(x) = (x-x_k)_+^3 - \frac{(x_p-x_k)}{d} (x-x_{p-1})_+^3 + \frac{(x_{p-1}-x_k)}{d} (x-x_p)_+^3$, $d = x_p - x_{p-1}$ and $x_+ = x$ if $x > 0$, 0 otherwise.

Knots are at x_k ($k=1, 2, \dots, p$). With only two knots, the formula is $y = \mathbf{a} + \mathbf{b}x$, a straight line. For three knots, the spline has three parameters, like a quadratic, but a natural spline is more useful in practice because its forecasts are linear, whereas forecasts based on quadratics tend to overshoot or undershoot data.

With $p=3$, the formula is $y = \mathbf{a} + \mathbf{b}x + \mathbf{c}_1 s_1(x)$, where, after a little algebra, we get

$$s_1(x) = x_1 \{ (x_2^2 + x_2 x_3 + x_3^2) - x_2 x_3 (x_2 + x_3) - x_1^3 \} + 3 \mathbf{c}_1 (x_2 - x_1)(x_3 - x_1)x \text{ for } x > x_3.$$

This tells us that before the first knot (x_1), y would increase at rate \mathbf{b} , whereas after the last knot (x_3), y would increase at rate $\mathbf{b} + 3 \mathbf{c}_1 (x_2 - x_1)(x_3 - x_1)$. So the increase in slope over the range of the data ($x_3 - x_1$) is $3 \mathbf{c}_1 (x_2 - x_1)(x_3 - x_1)$.

Consequently, the average **acceleration** over the data range is $3 \mathbf{c}_1 (x_2 - x_1)$.

For 21 years of data in the range $(0, 2.1)$ decades, a natural cubic spline with three equispaced knots has $x_1 = 0$, $x_2 = 1.05$ and $x_3 = 2.1$, so the acceleration for LST in degrees Centigrade per decade squared is thus $(3 \times 1.05)\mathbf{c}$, namely, $3.15 \times \mathbf{c}$.

Note that this differs from the formula given on Slide 6 in Session 3, which seems to be incorrect.

We can do a **simulation study** to check the formula, as follows.

Note that a simulation study starts with an assumption about a population, and then takes a random unbiased sample from this population with the objective of comparing estimates of specific population parameters with their known values.

For time series data, we assume that after adjusting for seasonal patterns and autocorrelation, observed data \mathbf{y} are determined by an additive model expressed as $\mathbf{y} = \mathbf{S} + \mathbf{z}$, where \mathbf{S} is a signal with known functional form and \mathbf{z} is a sequence of independent and identically distributed normal random variables with mean 0 and constant standard deviation (*white noise*).

Let's assume that the signal for season-adjusted day LST in a sub-region of interest follows a symmetric quadratic function over the 21-year range with maximum 10.5°C after 10.5 years and minimum values 0°C at the beginning and end of this period. Also assume that the noise has standard deviation 1°C . Here's some code to simulate data observed at 8-day intervals (46 per year).

```
set.seed(12345) # ensure repeatability
x <- c(1:(46*32))/460; nObs <- 46*21 # 21 (observed) + 11 future years
z <- rnorm(nObs,0,1) # white noise with sd=1
S <- 10.5-(x[1:nObs]-10.5)^2/1.05 # quadratic signal
y <- S+z # data observed
y <- c(y,rep(NA,46*11)) # unknown future values
kn <- 2.1*c(0:2)/2; p <- length(kn) # three equispaced knots
yy <- as.data.frame(cbind(y,x)) # database table
names(yy) <- c("y", "x") # variable names
d1 <- kn[p]-kn[p-1] # gap between last two knots
for (k in c(1:(p-2))) { # create spline function
  sk <- ifelse(x>kn[k],(x-kn[k])^3,0)
  sk <- sk-((kn[p]-kn[k])/d1)*ifelse(x>kn[p-1],(x-kn[p-1])^3,0)
  sk <- sk+((kn[p-1]-kn[k])/d1)*ifelse(x>kn[p],(x-kn[p])^3,0)
  yy[, (k+2)] <- sk
  names(yy)[k+2] <- paste("s", k, sep="")
}
mod2 <- lm(data=yy, y~.) # fit linear model
summary(mod2) # display results
```

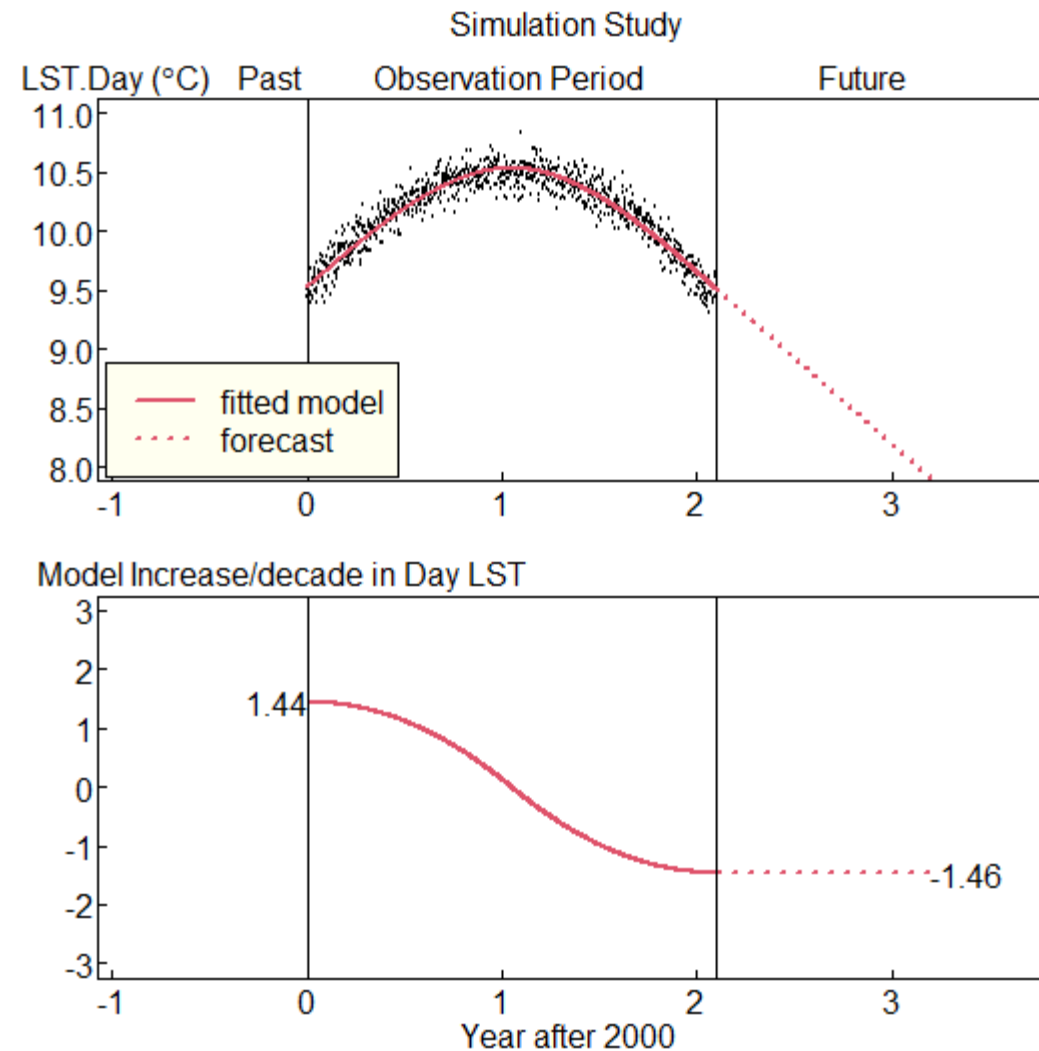
Given that $p=3$, this loop can be simplified to

```
sk <- ifelse(x>kn[1],(x-kn[1])^3,0)-
  ((kn[3]-kn[1])/d1)*ifelse(x>kn[2],(x-kn[2])^3,0)+
  ((kn[2]-kn[1])/d1)*ifelse(x>kn[3],(x-kn[3])^3,0)
```

These plots show results.

Estimated values are $b = 1.44$ and $c_1 = -0.439$, corresponding to initial increase in day LST $^{\circ}\text{C}$ per decade and acceleration $3 \times (-0.439) \times 1.05 = -1.38$ $^{\circ}\text{C}$ per decade² over the data range.

This acceleration corresponds to a decrease in slope of 1.38×2.1 ($2.9^{\circ}\text{C}/\text{decade}^2$) for the 21-year period of observation, which matches the decrease from 1.44 to -1.46 shown in the lower plot panel.



So the simulation study confirms the result shown on Slide 2 and shows that the formula given on Slide 6 in Session 3 is wrong.

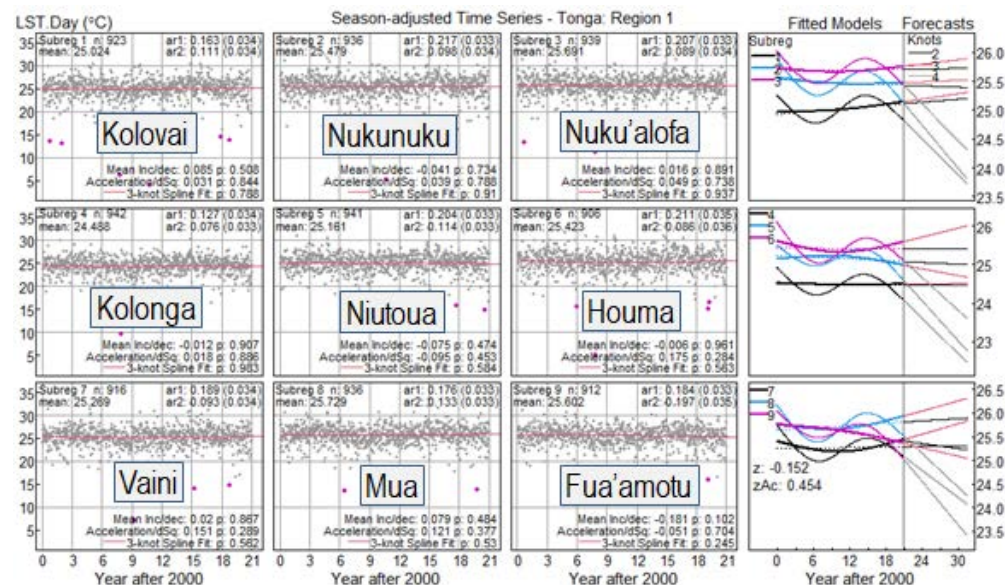
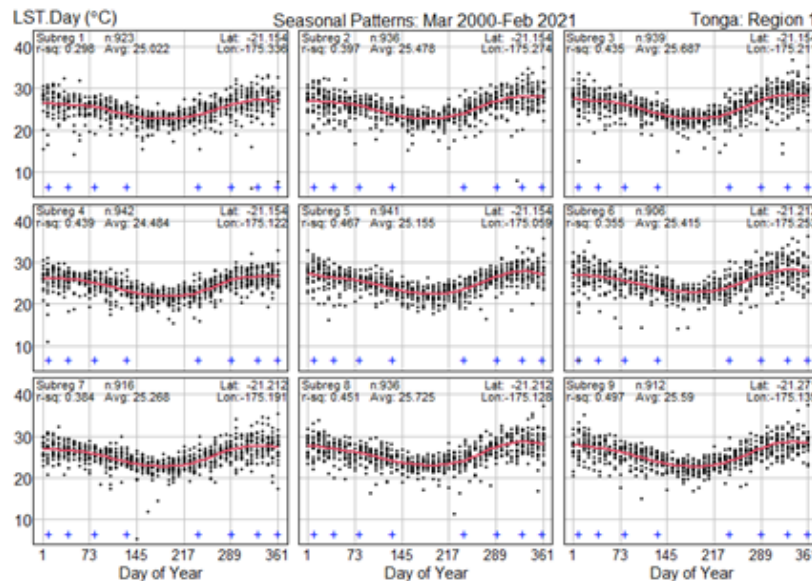
You can run the following code in [simulateTD.Rcm](#) to do this simulation.

```
windows(6,6)
par(mfrow=c(2,1), las=1, mar=c(1,0.5,2,0.3), oma=c(1.5,3,2,1), tcl=0.2, mgp=c(1.1,0.1,0))
plot(x, y, cex=0.2, xlim=c(-0.9,3.6), ylim=c(8,11), xlab="", ylab="")
ylab <- expression(paste(" LST.Day (", degree, "C", sep=""))
mtext(side=3, line=-0.1, adj=-0.1, ylab)
mtext(side=3, line=0.1, adj=0.16, "Past")
mtext(side=3, line=0.1, adj=0.41, "Observation Period")
mtext(side=3, line=0.1, adj=0.84, "Future")
mtext(outer=T, line=-0.5, adj=0.5, "Simulation Study")
abline(v=c(0,2.1))
points(x[1:nObs], fv[1:nObs], type="l", col=2, lwd=2)
fSp <- 1:(46*11) # future span
points(x[nObs+fSp], fv[nObs+fSp], type="l", col=2, lty=3, lwd=2)
nf <- length(fv); np <- length(fSp)
df <- 460*(fv[-1]-fv[-nf]) # model increase/decade
lg <- c("fitted model", "forecast")
legend("bottomleft", inset=c(0.01,0.01), lg, lwd=2, col=2, lty=c(1,3), bg="ivory")
plot(NA, xlim=c(-0.9,3.6), ylim=c(-3,3), xlab="", ylab="")
points(x[-1][1:nObs-1], df[1:nObs-1], type="l", col=2, lwd=2)
points(x[-1][nObs-1+fSp], df[nObs-1+fSp], type="l", col=2, lty=3, lwd=2)
mtext(side=3, line=0.1, adj=-0.13, "Model Increase/decade in Day LST")
mtext(side=1, line=1, adj=0.5, "Year after 2000")
abline(v=c(0,2.1))
text(0, df[1], round(df[1], 2), adj=c(1,0.5))
text(3.2, df[nObs], round(df[nObs], 2), adj=c(0,0.5))
```

Recall that in Session 3 we showed how a *computer function* `spp()` uses a *sinusoidal polar projection* to create maps that show what landforms on the Earth really look like from outer space.

Today we'll create another computer function `fitLST()` that fits natural cubic splines to LST trends adjusted for seasonal patterns and autocorrelation. We already have programs that create corresponding graphs, so we can do this by deleting the commands that create the graphs.

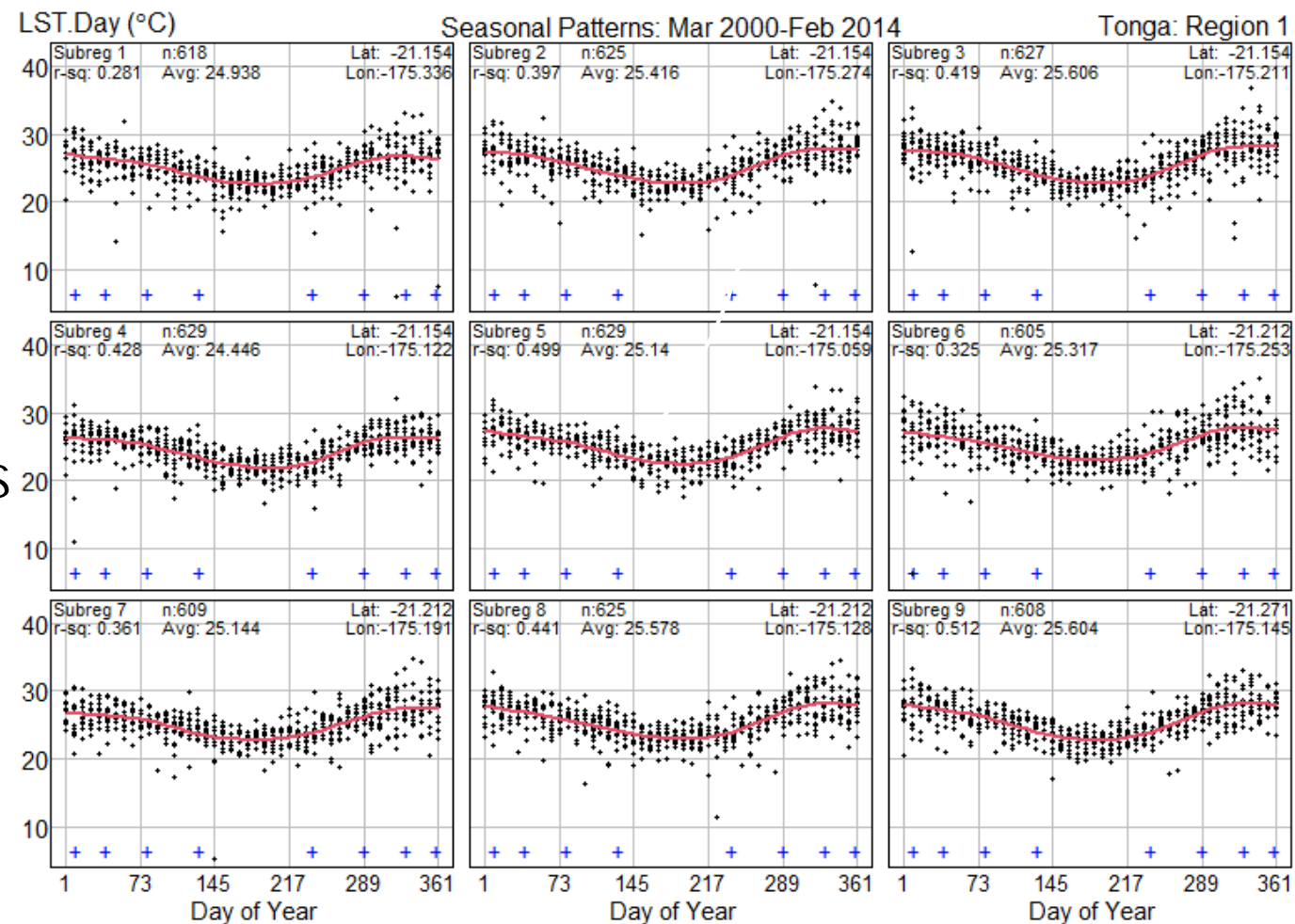
For example, in Session 3 a program `tgTD5b.Rcm` created seasonal patterns and time series for sub-regions in Tongakapu Island, as shown below.



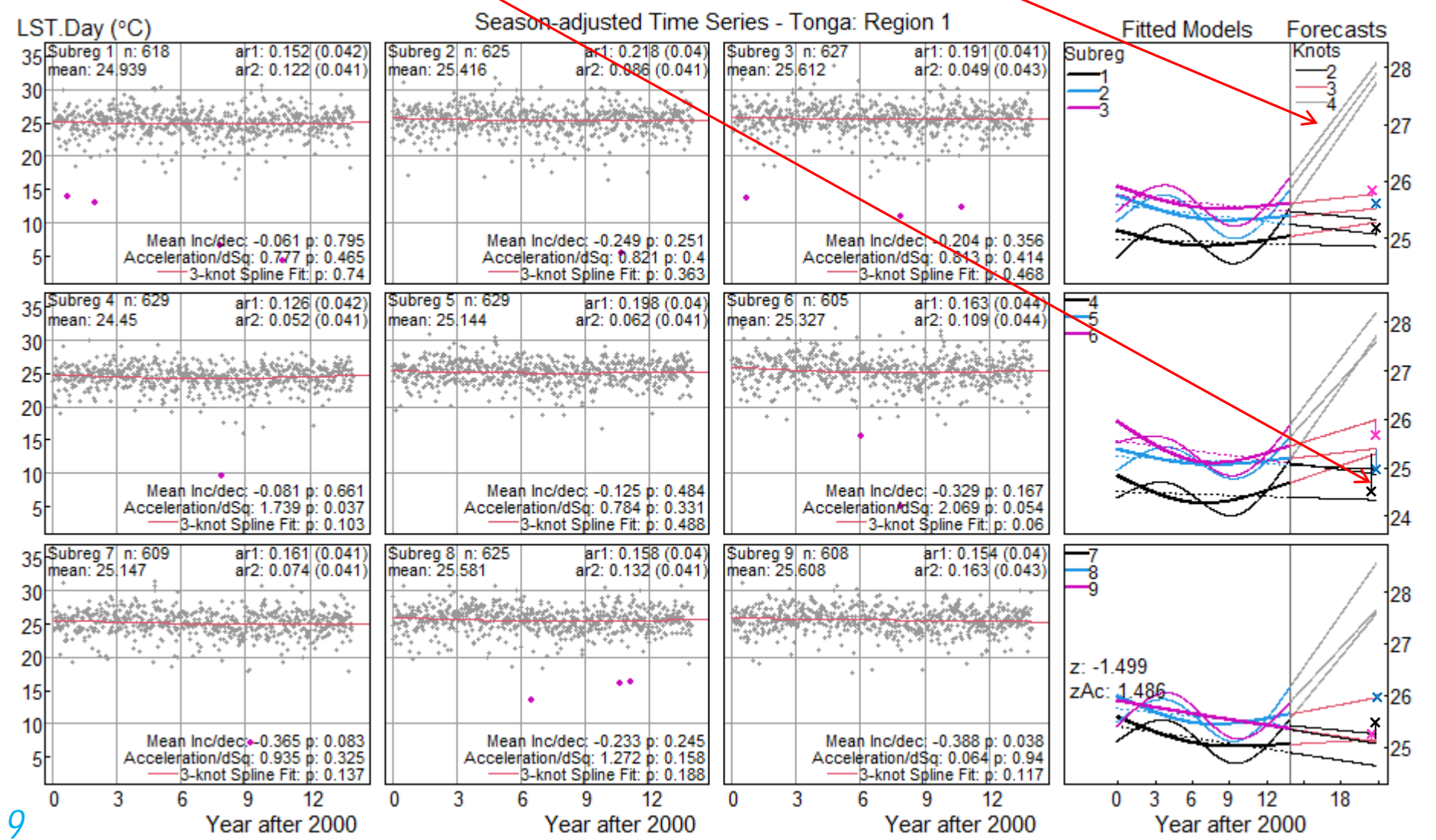
The program **tgTD5c.Rcm** does this. It has two further integer parameters (**yr1** and **yr2**) that specify the period of observed data, so if these values are 1 and 21, say, the data from day 49 in year 2000 to day 49 in year 2021 are selected, and the result is the same as what **tgTD5b.Rcm** gives. However, if **yr1** remains 1 and **yr2** is 14, the data extend from day 49 in year 2000 to day 49 in 2014.

This modification enables prediction of **known** fitted values, so forecasts can be checked.

Graphs shown here are seasonal patterns for Tongakapu sub-regions based on past data from 2000 to 2014.



Crosses show fitted values of 3-knot splines using data from 2008 to February 2021, compared with forecasts using data up to 2014. Forecasts for sub-regions are all reasonably accurate, all within 0.4°C except for sub-region 4 that differs from the fitted value by 1°C. But 4-knot splines badly over-forecast.



The function **fitLST.Rcm** applies code from **tgTD5c.Rcm** to sub-regions in other regions. Here's how it does that.

```
# fitLST.Rcm                                # fit model to LST data in (yr1,yr2) years inclusive
fitLST <- function(aa,sregs,yr1,yr2) {      # .. for region in place aa with sub-region IDs in sregs
  ff <- list.files()
  ffs <- ff[substr(ff,1,2)==aa & substr(ff,6,8)=="csv"]
  ffs <- ffs[as.integer(substr(ffs,3,4)) %in% sregs]
  nSubRegs <- length(sregs)
  nRegs <- floor(nSubRegs/9); subRegs <- sregs
  days <- 1+8*c(0:45)
  yds <- 2000000+100000*(yr1-1)+c(0:yr2)*1000
  T1 <- rep(days,2+(yr2-1))+rep(yds,each=46)
  T1 <- as.data.frame(T1)
  names(T1) <- "yrDay"
  T1$yrDay <- as.integer(T1$yrDay)
  yd1 <- 2000000+1000*(yr1-1)+49; yd2 <- 2000000+1000*yr2+57
  T1 <- subset(T1,yrDay>yd1 & yrDay<yd2)
  Lat <- NULL; Lon <- NULL; TD <- T1
  ..... (remaining code from tgTD5c.Rcm that creates data in ySA)
  ySA
}
```

Now that we can more easily create the fitted values using the `fitLST.Rcm` function, we can create graphs of fitted values with forecasts using the following program (`aTD6.Rcm`).

```
# aTD6.Rcm
# Analyse small samples of MODIS data from a specified place with natural cubic splines

rm(list=ls())                # remove redundant local variables

setwd("c:/world/lst_data")

aa <- "tg"                   # select Tonga
yr1 <- 1; yr2 <- 21          # period of observation (first and last years)

g1 <- c(1:9)                 # Tongakapu
g2 <- c(21:19,4:6,11:13)     # Ha'ano, Fotua, Lifuka, Tongakapu 1-3 + Ohonua
group <- 1
if (group==1) sregs <- g1
if (group==2) sregs <- g2

source("../fitLST.Rcm")
ySA <- fitLST(aa,sregs,yr1,yr2)
str(ySA)

yr1a <- 1; yr2a <- 14        # assume known data have shorter span
ySA1 <- fitLST(aa,sregs,yr1a,yr2a)
str(ySA1)
```

Fitted values and forecasts are now available in `ySA` and `ySA1`

We now have tools to make empirical forecasts for day LST data.

But the coloured crosses shown on the right side of the graph on Slide 9 were not put there by a computer program, but by examining the plots shown on the bottom right of Slide 7 and adding them as shapes using PowerPoint.

So instead we'll do it for the Tongakapu sub-regions by adding the following commands to **aTD6.Rcm**.

```
yr1a <- 1; yr2a <- 14                # assume known data have shorter span
source("../fitLST.Rcm")
ySA1 <- fitLST(aa,sregs,yr1a,yr2a)    # create data array ySA1
place <- ifelse(aa=="tg","Tonga","Unknown")
titl <- paste(place,"Region",group,"(Tongakapu)")
nObs=21*46
nReg <- round((ncol(ySA)-4)/36)
yFit <- ySA[(2+9*nReg+1):ncol(ySA)]
yFit1 <- ySA1[(2+9*nReg+1):ncol(ySA1)]
gp1Labs <- c("Kolovai","Nukunuku","Nuku'alofa","Kolonga","Niutoua","Houma",
             "Vaini","Mua","Fua'amotu")
```

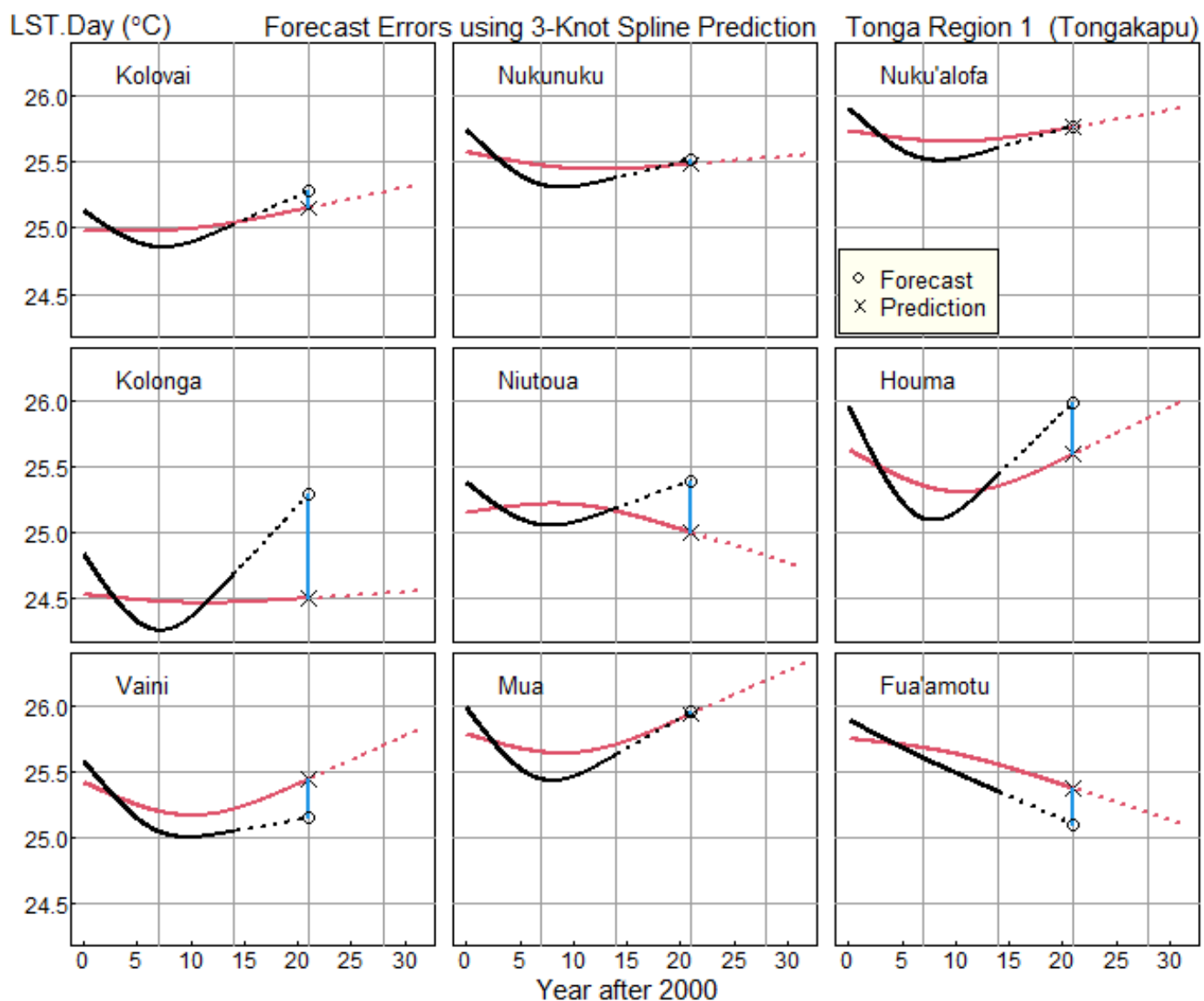
.....

The following further commands in **aTD6.Rcm** will create a graph of results.

```
windows(12,10)
par(mfrow=c(3,3),las=1,mar=c(0,0.5,0.5,0.3),oma=c(3,3,2,1),tcl=0.2,mgp=c(1.1,0.1,0))
yF <- yFit[,c(2,3*c(1:9))] # select 3-knot fits
yF1 <- yFit1[,c(2,3*c(1:9))]
ymin <- min(yF[,2:10]); ymin1 <- min(yF1[,2:10])
ymax <- max(yF[,2:10]); ymax1 <- max(yF1[,2:10])
ymin <- min(ymin,ymin1); ymax <- max(ymax,ymax1)
ylm <- c(ymin,ymax); xlm <- c(0,max(yF$t))
ylab <- expression(paste("LST.Day (" ,degree,"C)",sep=""))
for (j in c(1:9)) {
  plot(NA,xlim=xlm,ylim=ylm,type="l",xaxt="n",yaxt="n",xlab="",ylab="")
  abline(h=c(0:60)/2,col=8)
  if (j==1) mtext(side=3,line=-0.05,adj=-0.3,ylab,cex=0.9)
  if (j==2) mtext(side=3,line=0.1,adj=1,"Forecast Errors using 3-Knot Spline Prediction",cex=0.9)
  if (j==3) mtext(side=3,line=0.1,adj=1,titl,cex=0.9)
  if (j %in% c(1,4,7)) axis(side=2,cex.axis=1.1)
  if (j>6) axis(side=1,at=c(0:6)/2,lab=c(0:6)*5,cex.axis=1.1)
  if (j==8) axis(side=1,padj=1.4,at=1.51,lab="Year after 2000",tcl=0,cex.axis=1.4)
  abline(v=0.7*c(1:4),col=8)
  points(yF$t,yF[,j+1],type="l",col=2,lty=3,lwd=2)
  points(yF$t[1:966],yF[1:966,j+1],type="l",col=2,lwd=2)
  points(yF1$t,yF1[,j+1],type="l",lty=3,lwd=2)
  points(yF1$t[1:644],yF1[1:644,j+1],type="l",lwd=2)
  points(2.1,yF[966,j+1],pch=4,cex=1.6)
  points(c(2.1,2.1),c(yF[966,j+1],yF1[966,j+1]),type="l",col=2,lwd=2)
  points(2.1,yF1[966,j+1],cex=1.4)
  legend("topleft", bty="n",gp1Labs[j],cex=1.3)
  if (j==3) legend("bottomleft",inset=c(0.01,0.01),bg="ivory",leg=c("Forecast","Prediction"),pch=c(1,4),cex=1.2)
}
```


This graph appears when the program is executed. Blue vertical lines denote errors between predictions from fitting 3-knot splines to known outcomes and forecasts from extending 3-knot splines to data observed 7 years earlier.

Assuming data are homogeneous, the errors can now be used to create empirical bounds for future forecasts a further 7 years ahead after fitting a 3-knot spline model to data for the 14-year period from 2008 to 2021. This is our next task.



In this session we continued applying basic data analytic methods to samples of daytime land surface temperature remote sensing data reported from Earth-orbiting satellites from March 2000 to February 2021.

We focused on using a 3-knot natural cubic spline to fit daytime land surface temperature trends and to provide forecasts of future patterns up to 7 years ahead. We did this by using data 7 years earlier to predict the most recent 7 years and thus provide forecasting error bounds.

And we used computer functions to package analysis programs that simplify visual presentation of results.

These methods require extensive further assessment and improvement using global climate data available from NASA.

Please email me at don.mcneil@mq.edu.au if you'd like to work with us on this research topic.

Thank you for your patience. Hope to see you again before too long!