Linear and Nonlinear Correlation

Dr. Thomas Robey
Lynn Robey
Supercomputing Challenge Kickoff
October 1, 2023

R Statistical Software

R is a free statistical analysis software package available for Windows, Mac and Linux.

Binary distributions are available for Windows and Mac OS X at https://cran.r-project.org/. Many Linux distributions have R available in the package management systems or check the link above.

Jupyter notebooks (Julia - Python - R) can also be set up with the R kernel. Google Colab which is online Jupyter notebooks also can be used with R. The appendices have instructions for programming in R using Google Colab.

Pearson Correlation

The Pearson correlation evaluates the relationship between two continuous variables. The correlation can range between -1 and 1 with values near 0 indicating the two variables are not related, near -1 indicating they are inversely related and near 1 that they are positively correlated.

For more information: https://statisticsbyjim.com/basics/correlations/

Spearman Rank Correlation

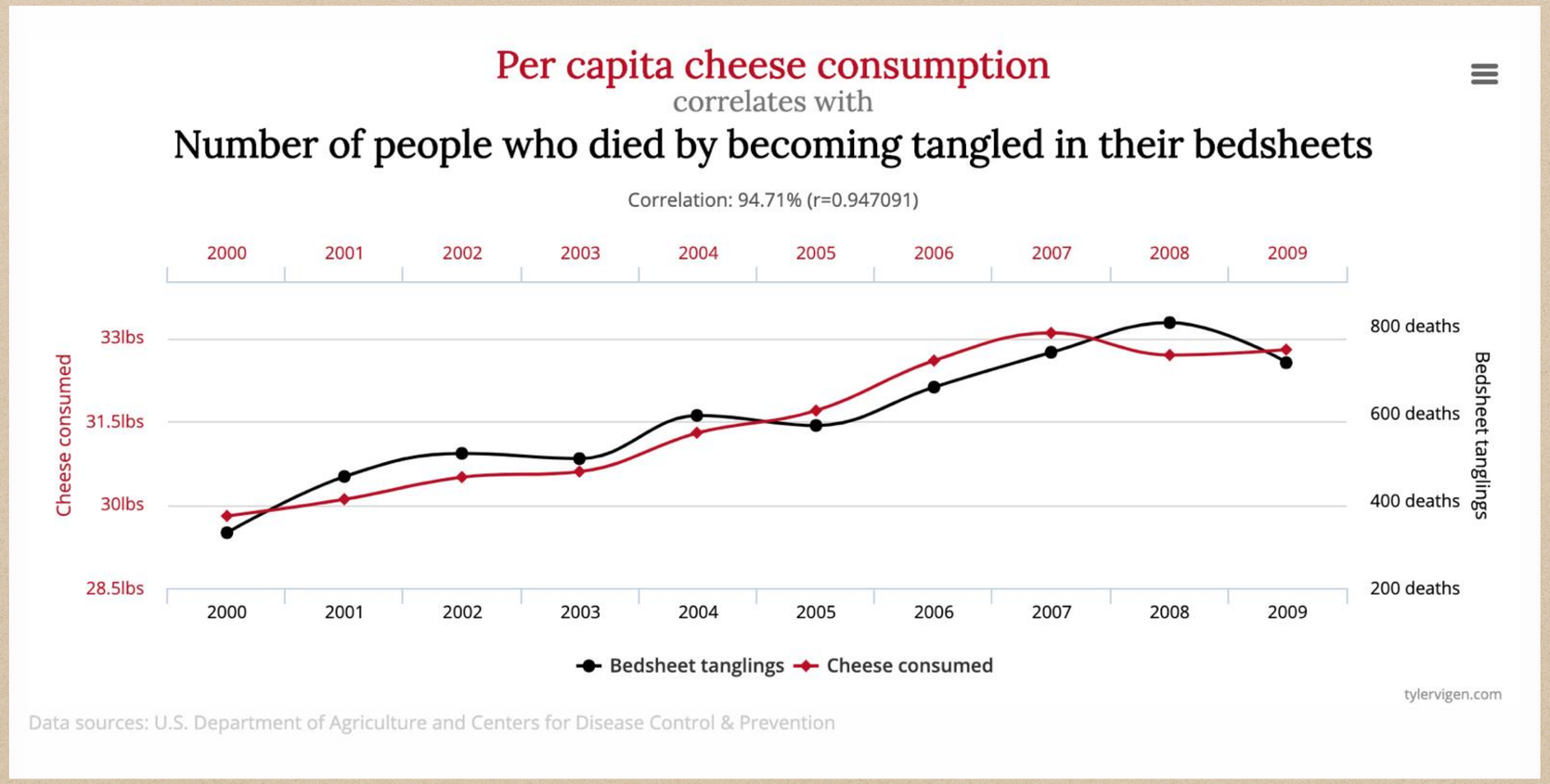
Many variables do not have a linear relationship. The Spearman rank correlation can be used for variables that have a nonlinear relationship that is monotonic. Monotonic means they must have an increasing or decreasing relationship.

The Spearman rank correlation replaces the data for each variable with their ranks and then computes the correlation.

For more information: https://statisticsbyjim.com/basics/spearmans-correlation/

Correlation and Causation

Correlation does not imply causation.

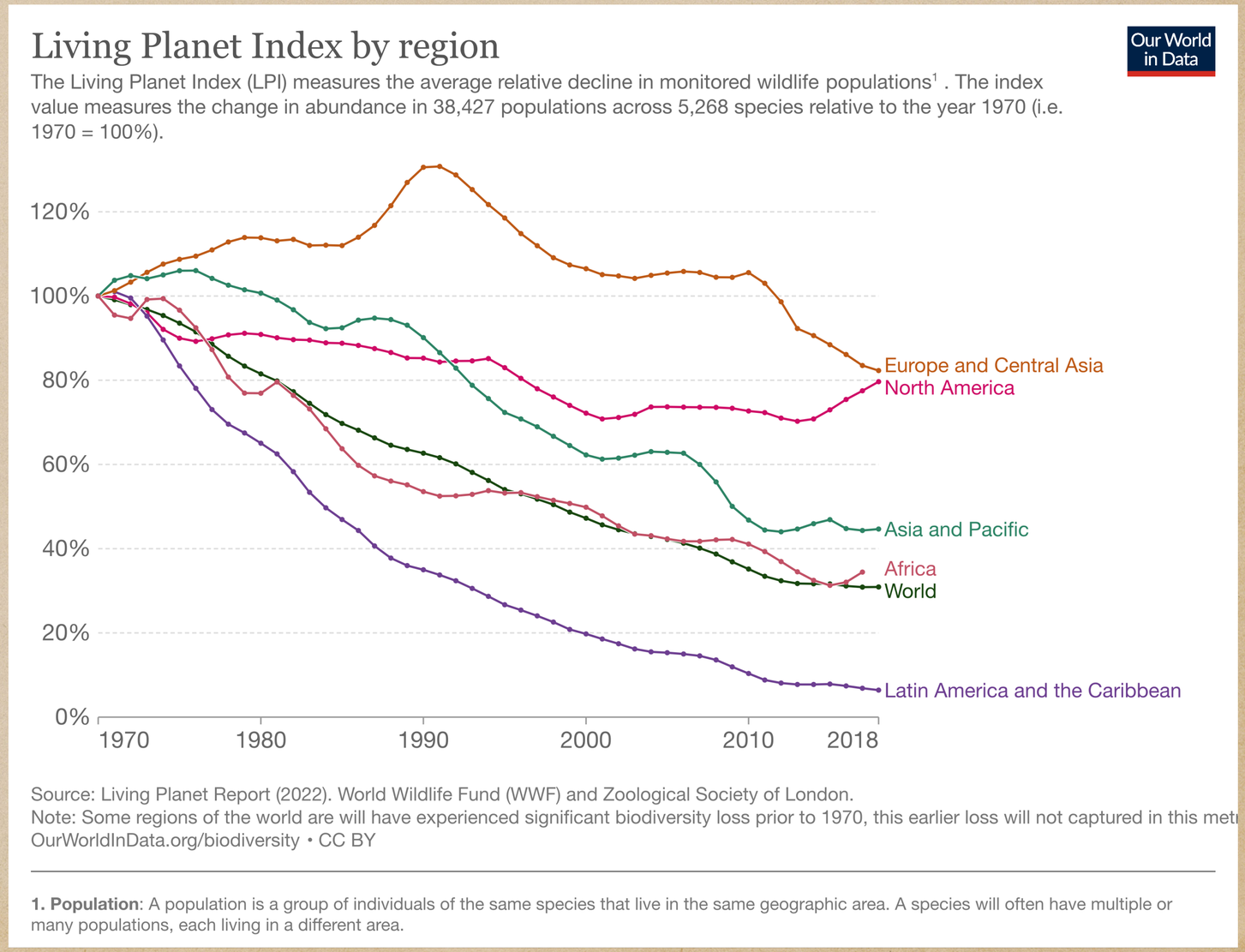


https://www.freecodecamp.org/news/why-correlation-does-not-imply-causation-the-meaning-of-this-common-saying-in-statistics/

Data Set

This session will use Living Planet data from Our World in Data:

https://ourworldindata.org/grapher/living-planet-index-by-region



To make it easy for this session the data is placed on the internet but you may need to download it from Our World in Data and read it in from a local file.

```
living_planet <- read.csv('https://gaiaes.com/living-planet-index-by-region.csv')
latin_america <- living_planet[living_planet$Entity == 'Latin America and the Caribbean', ]
latin_america
```

```
living_planet <- read.csv('https://gaiaes.com/living-planet-index-by-region.csv')
latin_america <- living_planet[living_planet$Entity == 'Latin America and the Caribbean', ]
latin_america
```

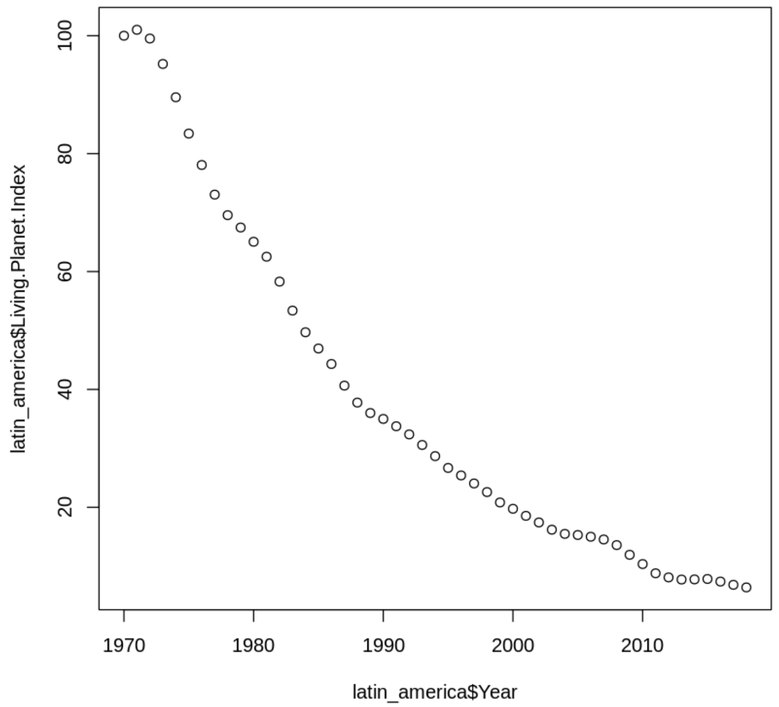
A data.frame: 49 × 4

	Entity	Code	Year	Living.Planet.Index
	<chr>	<chr>	<int>	<dbl>
196	Latin America and the Caribbean		1970	100.000000
197	Latin America and the Caribbean		1971	101.007160
198	Latin America and the Caribbean		1972	99.523110
199	Latin America and the Caribbean		1973	95.212230
200	Latin America and the Caribbean		1974	89.556350
201	Latin America and the Caribbean		1975	83.397830
202	Latin America and the Caribbean		1976	78.072804
203	Latin America and the Caribbean		1977	73.043780
204	Latin America and the Caribbean		1978	69.559350
205	Latin America and the Caribbean		1979	67.465335
206	Latin America and the Caribbean		1980	65.049887
207	Latin America and the Caribbean		1981	62.513804
208	Latin America and the Caribbean		1982	58.292377
209	Latin America and the Caribbean		1983	53.373235
210	Latin America and the Caribbean		1984	49.697760
211	Latin America and the Caribbean		1985	46.941832
212	Latin America and the Caribbean		1986	44.316828
213	Latin America and the Caribbean		1987	40.647778

Plot the data.

plot(latin_america$Year, latin_america$Living.Planet.Index)

```
plot(latin_america$Year, latin_america$Living.Planet.Index)
```



Compute the Pearson Correlation

cor.test(latin_america$Year, latin_america$Living.Planet.Index, method = 'pearson')

```
cor.test(latin_america$Year, latin_america$Living.Planet.Index, method = 'pearson')
```

```
        Pearson's product-moment correlation

data:  latin_america$Year and latin_america$Living.Planet.Index
t = -20.989, df = 47, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.9719666 -0.9135762
sample estimates:
       cor
-0.9505752
```

The Pearson correlation is -0.951 which indicates a strong negative correlation. The p-value is less than 0.05 which indicates it is statistically significant.

Compute the Spearman Rank Correlation

cor.test(latin_america$Year, latin_america$Living.Planet.Index, method = 'spearman')

```
cor.test(latin_america$Year, latin_america$Living.Planet.Index, method = 'spearman')
```

```
        Spearman's rank correlation rho

data:  latin_america$Year and latin_america$Living.Planet.Index
S = 39190, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
       rho
-0.9994898
```

The correlation is -0.999 which indicates a strong negative correlation. The p-value is below 0.05 which indicates the correlation is statistically significant.

Data Transformations

Computing the linear and nonlinear correlations is relatively straightforward. There are more approaches that involve the knowledge of the physical problem and insight. This is an example of such a method that uses data transformation.

The plot of the data for we have been considering looks like it may be an exponential decay. Population growth is typically exponential in the absence of any constraints. This data shows a population decrease so the proper model is not so clear but may be worth a try. Since

$\log(e^x) = x$

we take the log of the data.

cor.test(latin_america$Year, log(latin_america$Living.Planet.Index), method = 'pearson')

```
cor.test(latin_america$Year, log(latin_america$Living.Planet.Index), method = 'pearson')
```

```
        Pearson's product-moment correlation

data:  latin_america$Year and log(latin_america$Living.Planet.Index)
t = -98.009, df = 47, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.9986317 -0.9956596
sample estimates:
       cor
-0.9975625
```

So the Spearman rank correlation implied a nonlinear relation and we guess the model is exponential. Then by applying a log transformation of the data we arrive at a higher linear correlation.

Plot the linear model

```
linear_model <- lm(Living.Planet.Index ~Year, latin_america); linear_model
plot(latin_america$Year, latin_america$Living.Planet.Index)
abline(linear_model)
```

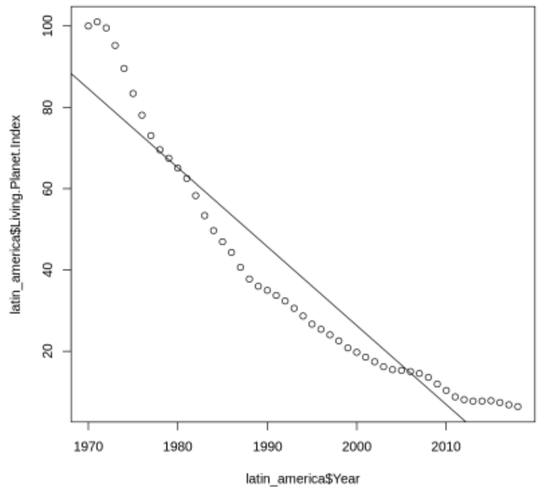
```
linear_model <- lm(Living.Planet.Index ~Year, latin_america); linear_model
plot(latin_america$Year, latin_america$Living.Planet.Index)
abline(linear_model)
```

Run cell (⌘/Ctrl+Enter)
cell executed since last change

executed by Thomas Robey
11:46 AM (5 minutes ago)
executed in 0.295s

```
Call:
lm(formula = Living.Planet.Index ~ Year, data = latin_america)

Coefficients:
(Intercept)         Year
   3909.468       -1.942
```



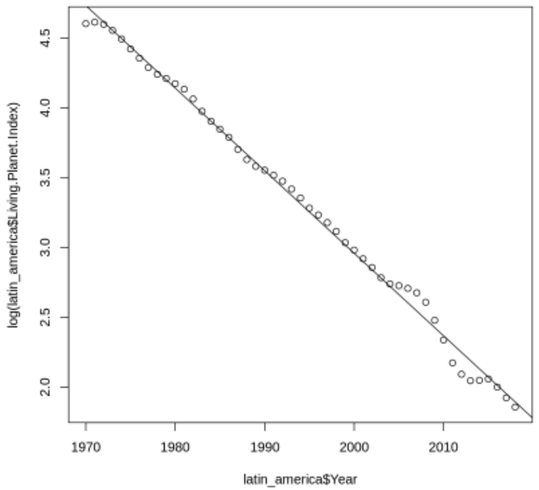
Plot the model after the data transformation.

transformed_model <- lm(log(Living.Planet.Index) ~ Year, latin_america); transformed_model
plot(latin_america$Year, log(latin_america$Living.Planet.Index))
abline(transformed_model)

```
transformed_model <- lm(log(Living.Planet.Index) ~ Year, latin_america); transformed_model
plot(latin_america$Year, log(latin_america$Living.Planet.Index))
abline(transformed_model)
```

```
Call:
lm(formula = log(Living.Planet.Index) ~ Year, data = latin_america)

Coefficients:
(Intercept)         Year
  121.24194     -0.05914
```

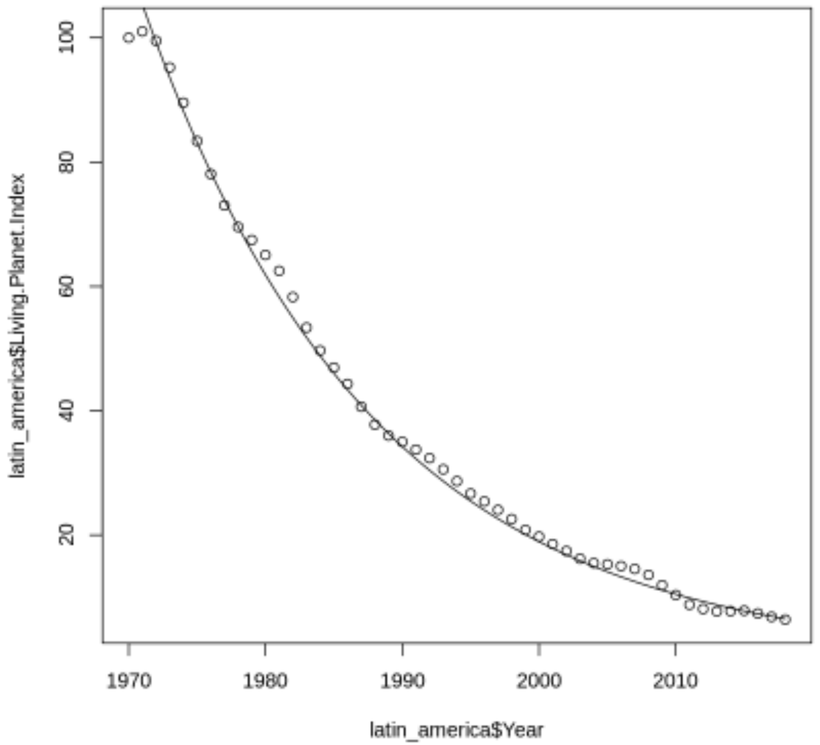


Reversing the transformation to the original data we have the function

$y = 4.5155x10^{52} * e^{-0.05915*x}$

```
x = seq(from = 1970, to = 2018, length.out = 2000)
func <- function(x) {
  exp(121.24194) * exp(-0.05915*x)
}
y = lapply(x, func)
plot(latin_america$Year, latin_america$Living.Planet.Index)
lines(x, y)
```

```r
x = seq(from = 1970, to = 2018, length.out = 2000)
func <- function(x) {
  exp(121.24194) * exp(-0.05915*x)
}
y = lapply(x, func)
plot(latin_america$Year, latin_america$Living.Planet.Index)
lines(x, y)
```



Likelihood-Ratio Test

The Likelihood-ratio test can be used to compare two models.

```
install.packages('lmtest')
library(lmtest)
lrtest(lm(Living.Planet.Index ~ Year, latin_america), lm(log(Living.Planet.Index) ~ Year, latin_america))
```

```
install.packages('lmtest')
library(lmtest)
lrtest(lm(Living.Planet.Index ~ Year, latin_america), lm(log(Living.Planet.Index) ~ Year, latin_america))
```

```
Installing package into '/usr/local/lib/R/site-library'
(as 'lib' is unspecified)

also installing the dependency 'zoo'


Loading required package: zoo


Attaching package: 'zoo'


The following objects are masked from 'package:base':

    as.Date, as.Date.numeric
```

A anova: 2 × 5

	#Df	LogLik	Df	Chisq	Pr(>Chisq)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	3	-177.02203	NA	NA	NA
2	3	69.56658	0	493.1772	0

The p-value is on the lower right of the table. Since it is lower than 0.05 the exponential fit is significantly better than the linear fit.

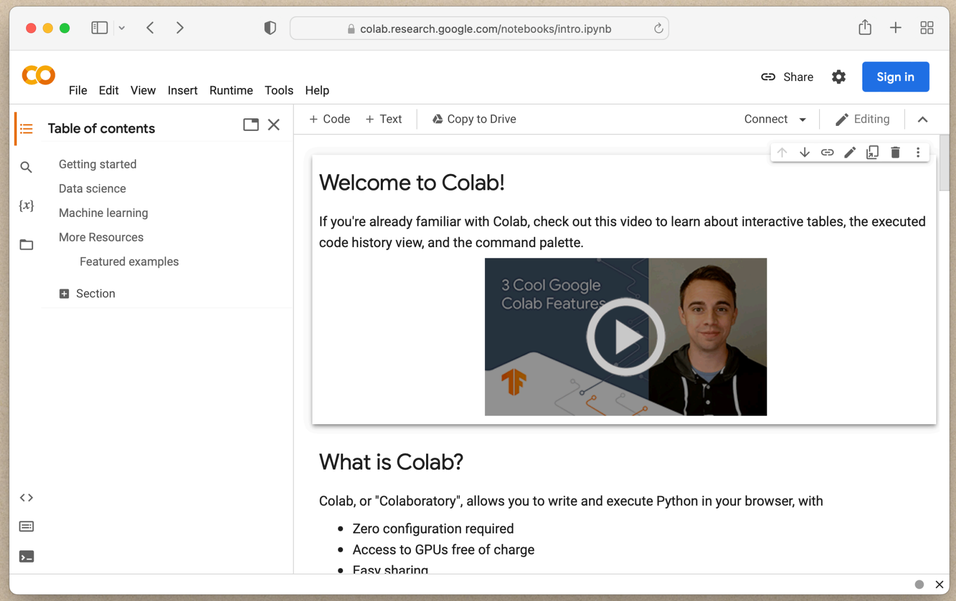
Exercises

Compute the Pearson and Spearman rank correlation for one of the other regions of the Living Planet Index. Does the data appear to be linear or nonlinear? Why?

Appendix I: Google Colab

Instructions for using Google Colab. Google Colab is an online version of Jupyter Notebooks. Go to

https://colab.research.google.com/notebooks/intro.ipynb



Click on the blue “sign in”button in the upper left corner and login to your Google account.

In the upper left corner click on File and New notebook.

Alternate approach

Go to https://accounts.google.com and log into your Google account. In the upper right corner click on the icon with 3 x 3 dots and click on Drive. Then click on + New, select More and then Google Colaboratory

Appendix II: R-notebook in Google Drive

To create a new R-notebook use the link:

https://colab.research.google.com/notebook#create=true&language=r

or the shorthand version

https://colab.to/r