Lab 5 Correlations and Linear Regressions

**There are 104 points within 6 questions in the lab**. Please make sure to save your code as ‘Lab5\_lastname.R’ and turn in a copy with all questions answered.

Introduction

In lecture, we have been discussing the idea of correlation. This is the idea that two things that we measure can be related to one another, either increasing or decreasing together in comparison to their means. For example, your personal happiness, which we could try to measure say with a questionnaire, might be related to other things in your life that we could also measure, such as number of close friends, yearly salary, how often you eat chocolate, or how many times you have said the word Nintendo in your life. Some of the relationships that we can measure are meaningful where the two items measured show a relationship due to an interaction between them. Some of the relationships are spurious, and do not reflect a meaningful relationship. In either case, correlations do not imply causation.

We have also learned that we can have a linear relationship between two variables where one variable is fixed (the x predictor variable) and is used to estimate the value of the response variable (y). In linear regressions, we have an idea that one variable may be directly influencing the response variable. For example, we may think that temperature influences fire size. We could first run a correlation between temperature and fire size and see if there is a linear relationship (correlation). We could then run a linear regression to see how much a change in temperature increases fire size (linear regression). Temperature is our fixed-predictor-independent variable and fire size is our random-response-dependent variable. Note that all three of these descriptions for the variables are used and are synonyms.

Back to the happiness example, we could try to predict a person’s happiness based on predictor variables, such as income or number of hours of sleep. We could also use spatial variables to predict happiness, such as distance to the ocean, distance to a trail head or public green space, or even latitude (are people closer to the equator happier than people farther away? i.e., does movement toward the poles predict a decrease in happiness?). Incorporating spatial predictor variables in our understanding of the world is at the heart of spatial statistics.

In this lab you will learn how to compute correlations between two variables in 'R' and ask some questions about the correlations that you observe. We will also continue our exploration of spatial data and calculate a spatial variable (latitude) and test the strength of a spatial variable in correlations and linear regressions.

Learning Goals

1. Compute Pearson’s r between two variables and with a spatial variable in R

2. Simple linear regression with R with two variables and with a spatial variable in R.

3. Discuss the possible meaning of correlations that you observe and the interpretations of R for a correlation and R-squared for a linear regression.

Data

We use data from the World Happiness Report (https://worldhappiness.report/). The World Happiness Report is a landmark survey of the state of global happiness that ranks 166 countries by how happy their citizens perceive themselves to be. A csv of the data for the year 2021 is included in the lab materials. The data includes annual survey results from the year 2008 to 2021 for each of the 166 countries. Beware that there are missing values ('NA') for some of the columns.

You should be able to see that there is data for many different countries, across a few different years.

There are lots of different kinds of measures, and each are given a name that should explain themselves.

You can find out more info here: <https://worldhappiness.report/faq/>

One of the most interesting columns is the ``Life.Ladder'' variable. This is a descriptor of life satisfaction and happiness. From the WHR website: ``This is called the Cantril ladder: it asks respondents to think of a ladder, with the best possible life for them being a 10, and the worst possible life being a 0. They are then asked to rate their own current lives on that 0 to 10 scale.'' Then WHR calculates the mean rating for each country and that's the value you see in ``Life.Ladder''.

There are six other variables to measure things that might influence happiness: income, healthy life expectancy, having someone to count on in times of trouble, generosity, freedom and trust, with the latter measured by the absence of corruption in business and government.

To start, let's explore the changes of the Life.Ladder over the years for the United States.

Follow the instructions in the R script. We will create a scatter plot of US happiness measured in annual average life ladder over time.

Then we will look at two countries together in a plot.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Before starting lab 5, let’s also take a moment and explore the world of **tidyverse**.

Tidyverse is a collection of R packages that are designed to work together seamlessly for data science tasks. It provides a unified and consistent framework for data manipulation, visualization, modeling, and reporting. Ggplot2, which we have already been using is core package in tidyverse. Tidyverse is becoming increasingly popular due to its simplicity an ease of use.

In addition to the unique functions in tidyverse that simplify data manipulation, tidyverse offers piping. This allows you to chain together multiple operations in a sequence. You can take the output of one function and pass it directly to the next function.

For example,

data %>%

filter(column1 > 5) %>%

group\_by(column2) %>%

summarize(meanValue = mean(column1))

"filter" is a useful function to exclude rows in a dataset based on criteria.

“group\_by” is function for grouping data into categories to summarize data

"summarise" allows you to calculate summary statistics of different rows/columns in a dataset

Other common functions include:

"select" is a useful function to get particular columns from a dataset

"left\_join" and it's cousins "full\_join", "inner\_join", "outer\_join", and "right\_join" allow you to merge two different spreadsheets

iris an example R dataset that people often work with to practice things. You can paste these lines of code into the R console and test the differences.

summary(iris)

### Filtering rows

## You might "filter" the iris data frame to get only the species "setosa" using the following using the base R equivalent to filtering rows. Square brackets imply reducing the number of rows/columns in a dataset

#base R

(setosaOnly <- iris[iris$Species == "setosa",])

Now, in tidyverse, a **%>%** symbol is called a "pipe", this tells tidyverse to "keep going to the next line". The code below does the same thing as the above code using simplified tidyverse language.

## tidyverse

(setosaOnly <- iris %>%

filter(Species == "setosa"))

Selecting columns. Maybe you only care about two of the columns -- "species" and "Sepal.Length"

Base R uses square brackets imply reducing the number of rows/columns in a dataset

equivalent to selecting columns.

(colSubset <- iris[, c("Species", "Sepal.Width")])

## tidyverse select functions

(colSubset <- iris %>%

select(c(Species, Sepal.Width)))

There are often equivalent ways to do things with complex base R coding, but tidyverse

is typically faster, easier to understand, and works better when combining multiple

steps in data wrangling. For example, we might want to filter to setosa and virginica,

select columns for "Species" and "Sepal.Width", and then calculate the mean "Sepal.Width"

for each of the two species. This can all be done using the following...

## Chaining multiple functions

## Base R - line one subsets the columns and rows to create a new object, line two calculates the mean "Sepal.Width" by species

(selectColSub <- iris[iris$Species == "setosa" | iris$Species == "virginica",

c("Species","Sepal.Width")])

## the above code first filters the rows within columns, and then selects the two columns.

(meanSepalw <- aggregate(selectColSub $Sepal.Width, by = list(selectColSub $Species), FUN=mean))

## the above code uses the result of the first command as input and tells R to calculate the mean for sepal width by species.

**## tidyverse version**

(meanSepalw <- iris %>%

filter(Species == "setosa" | Species == "virginica") %>% ## filter to the correct rows

select(c(Species, Sepal.Width)) %>% ## select the desired columns

group\_by(Species) %>% ## group your next statistics by Species

summarise(meanSepal = mean(Sepal.Width))) ## calculate summary statistics

In the tidyverse summarise function, you tell summarise what to name the resulting field with your statistics. This command above made a column called meanSepal.

In the lab we will do things in base R and tidyverse. You are not expected to be an expert in tidyverse in this course. It is simply helpful to know that there are multiple ways to do things. When you search for help on the internet, you should be able to distinguish if the help is in base R or tidyverse.

##\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Now back to the topic of happiness data.

**Step1** \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Reading in data and select countries to evaluate.

Follow instruction in lab to set your working directory and read in data.

We will first explore our data. it is important to take a look at your fields and see what is available in your data. You can also find the unique values of fields. Since we are working with countries, but not all countries, we want to know which countries are available in our dataset. We will do this in base R and then in tidyverse. Remember that by putting commands in parentheses, you print the result in the console.

## find unique countries in happiness database and print

## the unique function is useful for seeing all unique elements in a column.

(unique(happinessData$country))

## find unique countries in tidyverse

(happinessData %>%

distinct(country))

Note that in these commands, we did not make a new variable (i.e., we did not use the assignment command <-).

When we do not assign the manipulation to a new variable, the original variable does not change.

When you are exploring data, it is good to just print the results of your exploration and not always create new variables.

Next we will subset the rows to the country of interest.

## make subset of columns for a select country in base R selecting appropriate rows and columns

happinessDataUS <- happinessData[which(happinessData$country== 'United States'),]

#check your work

dim(happinessDataUS)

head(happinessDataUS)

## you now have just the US happiness data and all the associated columns

## in tidyverse - you get the same answer using the filter function

happinessDataUS <- happinessData %>% #filter to the US

filter(country == 'United States')

dim(happinessDataUS)

head(happinessDataUS)

In these manipulations, we assigned a new variable happinessDataUS to the result. Now we can use this new variable for other functions. We will map the scatter plot.

## plot US happiness Life ladder over time in a scatter plot using base R

plot(x=happinessDataUS$year, y=happinessDataUS$Life.Ladder, main = "US Mean Life.Ladder, 2008 - 2020")

# Plot the scatter plot with ggplot

ggplot(happinessDataUS) + aes(x=year, y=Life.Ladder) + geom\_point()+

ggtitle("US Mean Life.Ladder, 2008 - 2020")

Answer the questions in the lab.

**Step 2\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Next we will compare the happiness of two countries. In this next example, you can see how tidyverse makes the operations much simpler.

##Now let's look at two countries in base R

happinessDataGermany <- happinessData[which(happinessData$country== 'Germany'),]

## base R plot

plot(x=happinessDataGermany$year, y=happinessDataGermany$Life.Ladder, main = "Germany Mean Life.Ladder, 2008 - 2020", col='red', pch=19)

## add the next country as points to the first plot

points(x=happinessDataUS$year, y=happinessDataUS$Life.Ladder,col='blue', pch=19 )

## tidyverse and ggplot

happinessDataUSGermany <- happinessData %>% #filter to the US or Germany at same time

filter(country == 'United States' | country == 'Germany') %>%

select(country, Life.Ladder, year) #select the columns that you want to keep

ggplot(happinessDataUSGermany) + aes(x=year, y=Life.Ladder, color=country) + geom\_point() +

ggtitle("Mean Life.Ladder, 2008 - 2020")

Follow along in the lab for a few steps to make boxplots and summary statistics.

Note that creating summary statistics in tidyverse is more complicated than it is in base R.

**Step 3 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Correlations.**

After exploring the data, we will create correlation coefficients between variables.

Remember in a correlation, we get an R value that is between -1 and 1. This gives us the **direction** and the **strength** of a relationship between two variables.

Strength of Relationship: The absolute value of the correlation coefficient indicates how strong the relationship is. A correlation close to 1 (positive) or -1 (negative) indicates a strong relationship, while a correlation closer to 0 indicates a weak relationship.

Direction of Relationship:

Positive Correlation (r > 0): When one variable increases, the other tends to also increase. For example, as the number of hours studied increases, the exam score tends to increase.

Negative Correlation (r < 0): When one variable increases, the other tends to decrease. For example, as the number of cigarettes smoked per day increases, lung capacity tends to decrease.

No Correlation (r = 0): There is no linear relationship between the variables. This doesn't mean there is no relationship at all, just that it's not linear.

Answer the questions in your lab.

**Step 4 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Linear Modelling.**

A linear model is a fundamental statistical and mathematical tool used to describe the relationship between two variables. It assumes that this relationship can be adequately approximated by a linear equation, which is a straight line when visualized on a scatterplot. The basic idea is to fit a straight line to the data in such a way that it best represents the relationship between the variables.

A simple linear model expresses the relationship between two variables, usually denoted as "Y" (the dependent variable) and "X" (the independent variable), using a linear equation:

Y=β 0 +β 1X+ε.

Y is the dependent variable (the one you want to predict or explain).

X is the independent variable (the one you use to make predictions).

β 0 is the intercept, representing the value of Y when X is zero.

β 1 is the slope, indicating the change in Y for a one-unit change in X.

ε represents the error term, accounting for the variability that the model cannot explain.

To fit a linear model in R, you use the **lm()** function. Inside the function, you give it the two arguments to complete the formula with the variables you want to model:

**lm(y~x)**

Notice how the **y variable comes FIRST and the x comes SECOND**. You can interpret this as, response variable y is estimated by predictor variable X.

Let's fit a linear model predicting Life.Ladder (response variable) using Freedom.to.make.life.choices (predictor variable). See code in R.

The results of lm() What lm(y~x) estimate the coefficients for the intercept and slope. You may think of this equation as y = mx +b. m = slope and b = the intercept.

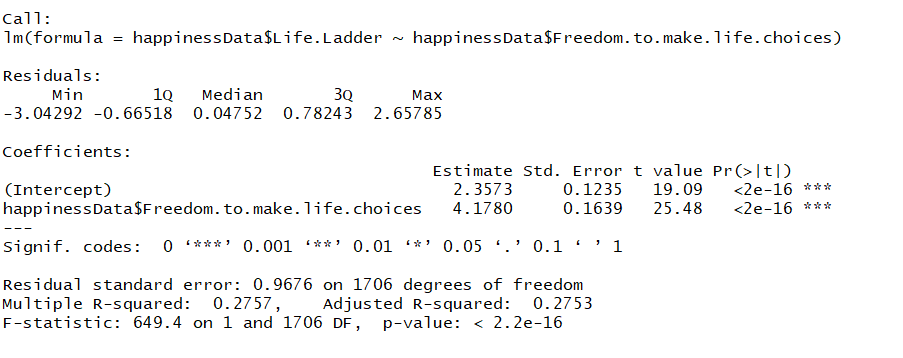
**Once the linear model is fitted, it can be used to make predictions**. Given a value of X, you can predict the corresponding Y by plugging it into the linear equation. Predictions are based on the slope (β1) and the intercept (β0). You can not use a correlation coefficient to make predictions.

The quality of the linear model is often assessed using metrics like the coefficient of determination (r2), which indicates how well the model explains the variance in the dependent variable.

**A strong linear relationship does not prove that changes in X cause changes in Y.**

The adjusted R-squared is a modified version of the R-squared. It is a measure of how well the independent variables in the model explain the variability of the dependent variable. The adjusted R-squared takes into account the number of independent variables in the model, providing a more accurate assessment of model fit. Since we are working with one or two variables, the adjusted R-squared will be the same as the R-squared.

We can summarize the results of the linear model. For now, let’s just focus on the R-squared value.



**The R-squared value gives us the proportion of variance in y predicted by X.** Higher R -squared values mean that the predictor variable is explaining more of the variation in Y than variables with lower R-squared values. In essence, a higher r-squared means that X is a strong predictor of Y.

We can also get the intercept and slope from the linear regression. The intercept is the amount of life ladder (y variable) a person has before accounting for freedom to make life choices (the x variable). The slope tells us how much life ladder increases for every increase in freedom to make life choices.

In this example, the slope of 4.17 tells us that for every one increase in freedom to make life choices, life ladder happiness increases 4 times.

Answer the questions in the lab.

**Step 5**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

For the second part for the lab, we will bring in spatial data.

We will use the world country boundaries to ask if life ladder happiness is predicted by distance from the equator? To get distance from the equator, we will use the absolute value of the latitude coordinate for countries reporting on happiness. You may hypothesize that life happiness is greater at the equator due to more day length.

In spatial models, using spatial variables is one of the first steps to understanding how geography matters. We could think of other variables as well that we could calculate in a GIS/R software program – like length of coastline. For our first spatial model, we will use the easily calculated latitude of the center point of each country.

In the R script you will read in the world countries shapefile. This is a polygon shapefile.

Then we need to join the happiness data to the country shapefile by a common field. Since both databases have a field that has country name data, we can use each country field and join the tables. The important piece is that the data in each of the fields must match. Since both databases, spell the country names similarly, there is a perfect match when a country has corresponding happiness data.

We will do these joins in base R and tidyverse.

In both base R and tidyverse, you need to specify which fields are used to join the databases.

##We can join the happinessData to the spatial data together by these column in Base R using the merge() function.

##we use by.x and by.y to tell the function which fields are the same in each database.

countriesH <- merge(countries, happinessData, by.x= "COUNTRY", by.y = "country", all.x = TRUE)

You can read more on the options for merge() here: <https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/merge>

In tidyverse, we will use function from the join family. There are many functions in tidyverse that join together databases: left\_join(), right\_join(), inner\_join(), full\_join(). We will use the left\_join(), which will act in the same way as all.x in base R. It will keep all of the X data and the y data that matches.

Read more on the tidyverse join family here: <https://tavareshugo.github.io/r-intro-tidyverse-gapminder/08-joins/index.html>

## Joining database in tidyverse, we use one of the join family functions. Note that since we use piping in tidyverse,

## tidyverse assumes that everything after the pipe uses the stuff before the pipe as the input.

countriesH <- countries %>%

left\_join(happinessData, by = c("COUNTRY" = "country"))

**Step 6** \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

In this final step, we will put all the pieces together. All the steps are within questions. You will find the latitude of each country point and then run a linear regression, predicting life ladder by latitude.

Nice work finishing this lab! You just completed your first spatial model!