

Statistics Labs for Psychology

Zakary A. Draper

2021-09-22

Contents

Lab Outline	5
Summary	5
Learning Outcomes	5
Software	5
Evaluation	6
Tentative Schedule	7
1 Introduction to R	9
1.1 Learning Objectives	9
1.2 Prepare for the Lab	9
1.3 Lab Activity	10
1.4 Discussion Questions	13
1.5 Resources	13
2 Introduction to R (Part 2)	15
2.1 Learning Objectives	15
2.2 Prepare for the Lab	16
2.3 Lab Activity	16
3 Data Visualization	19
3.1 Learning Objectives	20
3.2 Prepare for the Lab	20
3.3 Lab Activity	20

Lab Outline

Summary

The lab component of this course primarily concerns statistical methods and analysis, and of course the use of computer programs. The universe of statistical models is extraordinarily vast. We will focus on common statistical tests that can be broadly classified as linear models. Students will learn to model a diverse array of hypotheses as linear models.

Learning Outcomes

At the conclusion of this course, students will be able to:

1. Appropriately evaluate the strength of evidence provided by common statistical tests conducted using a linear regression framework.
2. Conduct common statistical tests in R.
3. Produce informative, aesthetically pleasing, data visualizations, in accordance with APA style guidelines.
4. Utilize online resources to expand their knowledge related to outcomes 1–3, such that they will be able to understand and appropriately utilize statistical concepts that are not taught in this course.

Software

R

R refers to both a programming language for statistical computing and software. It is a free and open source project. We will be doing all of our data analysis using the R programming language.

RStudio

RStudio is a commercial organization that offers a free and open source version of desktop software. The RStudio desktop software is an integrated development

environment. Which is to say, it is software for executing R code. RStudio has several benefits over the base R software.

Syzygy (Optional)

The R and RStudio software are available for Windows, Mac, and Linux operating systems. Neither will operate on a tablet or Chromebook. If you are using a tablet or Chromebook, you can run R in a web browser using a web application called Jupyter Notebook. A Jupyter Notebook that has been set up to run R is available to UBCO students at <https://ubc.syzygy.ca/>. You can sign in to Syzygy with your CWL.

Jupyter Notebook is an entirely separate interface from RStudio. The R code is the same, but the interface is different. We recommend using RStudio if possible, not because it is better, but because it is what we will be using. However, if RStudio is unavailable to you, or you just want to use Jupyter, then you should feel empowered to do so. You can absolutely be successful in this course using Jupyter.

XQuartz

If you are using a Mac, you will need to install XQuartz. XQuartz is required to install some of the R packages we will make frequent use of throughout the course. It is free and open source.

Zoom

Office hours with the lab instructors will be held via Zoom.

Canvas

We will use Canvas to share information and documents, including data sets, slides, and other useful materials.

Evaluation

The lab will be worth 30% of your final grade, divided across lab reports (21%), and quizzes (9%).

Lab Reports (21%)

Lab reports are brief, APA-style manuscripts, reporting and discussing the results of a study. You will receive 21% of your grade for completing 3 lab reports, each of which is worth 7% of your final grade. For each lab report, you will be provided with preregistration information for a hypothetical study as well as simulated data from that study. You will be responsible for writing the results and

discussion, creating an appropriate data visualization, and preparing a reproducible R script, which will be submitted with the report. Specific instructions for each lab are provided in the lab manual.

You may choose to complete lab reports on any of the statistical tests covered in the lab manual. Lab reports for a given lab are due the Monday following the lab at 11:59 PM. Late lab reports will not be accepted. Lab reports will be marked and returned with detailed feedback by the beginning of the next week's lab.

Optionally, you may submit a 4th lab report. If you do, your final lab report grade will be based on the best 3 of the 4 labs you submit.

Quizzes (9%)

There are two quizzes. The first is worth 3% of your final grade. It will cover R fundamentals that you will need to master to be successful in this course (and in 373). The second and final quiz is worth 6% of your final grade. It will take the form of a lab report that is completed during regular lab time.

Tentative Schedule

As indicated by the title of this section. This schedule is tentative and therefore subject may change. These are the topics we plan to cover, in the order we expect to cover them.

Labs 1–2 will cover the fundamentals of R and RStudio. Lab 3 will introduce you to data visualization using `ggplot2`. The next lab will be a quiz that will require practical application of the skills you learn in labs 1–3.

Beginning in week 6, each lab will cover one statistical test. For each of these statistical tests, you will be presented with preregistered information from a hypothetical study. Most of these studies are based on real research programs (many of which are being actively investigated by researchers at UBCO!). If the content of one of the labs interests you, consider reaching out to the professor whose research inspired it. There may be opportunities to work with that professor on a directed studies or honours project.

Week	Date	Topic
1	Sep. 09	Introduction to R
2	Sep. 16	Introduction to R, part 2
3	Sep. 23	Data visualisation with <code>ggplot2</code>
4	Sep. 30	National Day for Truth and Reconciliation. No Lab.
5	Oct. 07	Quiz 1
5	Oct. 14	One sample t test
6	Oct. 21	Paired samples t test
7	Oct. 28	Independent samples t test

Week	Date	Topic
8	Nov. 04	Correlation
10	Nov. 11	Midterm Break
11	Nov. 18	ANOVA
12	Nov. 25	Factorial ANOVA
13	Dec. 02	Final Quiz

Chapter 1

Introduction to R

1.1 Learning Objectives

After completing this lab, you should be able to:

- Explain what R is and describe its main uses.
- Give (at least) 3 benefits of using R for data analysis.
- Create R objects.
- Identify R objects of the following R classes: numeric, character, logical, data.frame.
- Apply basic operations to vectors in R.
- Explain the basic structure of function calls.
- Write scripts to manually enter data as `data.frames` in R.

1.2 Prepare for the Lab

Before coming to the lab, install R and RStudio on your computer. There are computers in the lab with R and RStudio installed, so if you are planning to use the lab computers for the semester, then you do not need to install

1.2.1 Install R

You can find links for downloading R and other information about R on the R website <https://www.r-project.org>. You can download R from one of several CRAN mirrors. The simplest option is the cloud (<https://cloud.r-project.org/>), which will automatically redirect you to a local server.

1.2.2 Install RStudio

Download and install the desktop version of RStudio from <https://rstudio.com/products/rstudio/>.

1.2.3 I'm Using a Tablet or Chromebook

You cannot install R or RStudio to these devices. You can run R using a Jupyter Notebook. The simplest way to make this happen is going to <https://ubc.syzygy.ca/> in your web browser and logging in with your CWL. Make sure you are able to sign in to this before the lab.

1.2.4 Can I Use a Lab Computer?

Yes. There are computers in the room where we will be holding the lab that have R installed already. If you are planning to use the lab computers for the entire semester, you are welcome to do so. Just know that will likely become fairly annoying when you have to work on lab work outside of regular lab hours (or if you are unable to attend the lab for some reason). Even if you plan on primarily using the lab computers, we recommend having a personal machine that you can use for doing your lab work.

If you do plan to use the lab computers, visit the lab room before the lab and ensure that you are able to sign in to the computers. We will not be able to take time out of the lab to provide support with logging into the campus machines.

1.3 Lab Activity

1.3.1 Mathematical Operations

Write R code to solve each of these equations:

1. $1/20$
2. $\frac{1}{20 \times 20}$
3. $1 - .95^{10}$
4. $\sqrt{19}$

1.3.2 Object Assignment

Use the assignment operator, `<-`, to define R objects with the names and values shown in the table below.

name	value
x	10
y	4.2
haiku	a haiku about R
excited	logical, honest answer to the question, "I am excited to learn R"

1.3.3 Object Classes

Use `class()` to identify the class(es) of each of the objects you just created.

1.3.4 Longer Vectors

A vector is a list of items of the same type. Vectors can contain numbers, characters, or logicals, so long as every item in that vector is the same class. The objects you have created thus far are all vectors of length one. That is, they each contain one item. But vectors more often contain many items.

You can combine items into a vector using `c()`, which is the concatenate/combine function. This function takes any number of arguments, each separated by a comma, and combines them into a vector.

Use the assignment operator, `<-`, and the concatenate function, `c()` to define R objects with the names and values shown in the table below.

name	value
dub	a numeric/double vector of length 3
dub2	a numeric/double vector of length 4

You can also name elements in your vector, as in the example below.

```
ages <- c("Declan" = 17, "Ava" = 19, "Liam" = 20, "Charlotte" = 19)
```

For longer function calls, it is easier to read if they are spread out over several lines. The code below is formatted differently, but functionally identical to the code which used only one line.

```
ages <- c(
  "Declan" = 17,
  "Ava" = 19,
  "Liam" = 20,
  "Charlotte" = 19
)
```

```
ages
```

```
##      Declan      Ava      Liam Charlotte
##         17         19         20         19
```

This can be useful, but a more common approach is to use data frames.

1.3.5 Data Frames

A `data.frame` is a special class of R object that groups vectors into columns. This is a useful way of organizing and presenting data. It is easy to read and

understand. The data we will work with this semester will be organized in a `data.frame`.

Create a `data.frame` using the function `data.frame()`. Like `c()`, `data.frame()`, combines its arguments. However, instead of combining elements into a vectors, `data.frame()` binds vectors into columns of a table.

```
data.frame(
  name = c("Declan", "Ava", "Liam", "Charlotte"),
  age = c(17, 19, 20, 19)
)
```

```
##      name age
## 1  Declan  17
## 2    Ava  19
## 3   Liam  20
## 4 Charlotte 19
```

This `data.frame` has the same information as the vector `ages` we created previously. Except the information is stored in two vectors—`name` and `age`—instead of one named vector. The ability to store information this way becomes increasingly useful if we want to more information about these individuals, as in the example below.

Recreate the following in R. Save the result as an object named `happiness`.

```
##      name age group shs_1 shs_2 shs_3
## 1  Declan  17     A     3     4     4
## 2    Ava  19     A     4     3     3
## 3   Liam  20     B     5     5     4
## 4 Charlotte 19     B     4     5     5
```

1.3.6 Add a Column to a Data Frame

Use `$` to add a column called `shs_total` to `happiness`, which is the sum of `shs_1`, `shs_2`, and `shs_3`. The result should look like this.

```
##      name age group shs_1 shs_2 shs_3 shs_total
## 1  Declan  17     A     3     4     4         11
## 2    Ava  19     A     4     3     3         10
## 3   Liam  20     B     5     5     4         14
## 4 Charlotte 19     B     4     5     5         14
```

1.3.7 Summary Statistics

Use R functions to compute the mean, median, and standard deviation of `shs_total`.

1.4 Discussion Questions

Be prepared to answer the following questions. It is okay (even good!) to use R to discover the answers to these questions.

1. What is the class of `happiness`? Why does it make sense to store data using objects of this class?
2. What is the class of `happiness$shs_tot`?
3. Why is the `$` needed to reference columns in a `data.frame`?
4. What is the output of `happiness$shs_tot + 2`?
5. What class is the object produced by executing: `c(happiness$shs_tot, happiness$group)`?
6. What is the difference between `1` and `"1"` in R?
7. Why does `Happiness$shs_tot + 2` return an error?

1.5 Resources

Slides

Chapter 2

Introduction to R (Part 2)

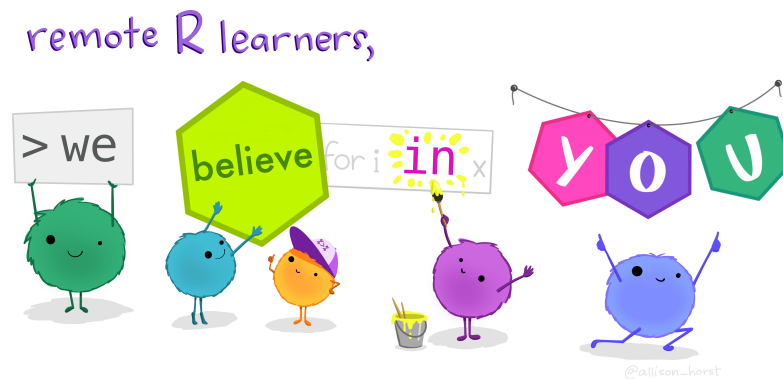


Figure 2.1: We believe in you!

2.1 Learning Objectives

After completing this lab, you should be able to:

- Reference files using absolute and relative paths.
- Create an R data frame from a data file stored on your computer.
- Convert numeric or character vectors to factors.
- Index vectors and data frames.

2.2 Prepare for the Lab

- Make sure you grasp the concepts from lab 1.
- Read Workflow: projects from R for Data Science.
- Read the following sections from Learning Statistics with R:
 - 4.6 Useful things to know about variables
 - 4.7 Factors

2.3 Lab Activity

Researchers at the Vancouver and Okanagan campuses of The University of British Columbia have conducted a survey of child sleep. The results of this survey are found in the file `child-sleep-data.csv`. The table below provides names and descriptions for each variable in the data set.

Name

Description

`id`

A randomly generated participant ID, unique to each participant.

`age`

The age of the child in the study.

`campus`

The campus from which the participant was recruited (“Point Grey” or “Okanagan”).

`sleep_location`

Where does the child sleep? 1 = co-sleeps with parent(s); 2 = sleeps in own room; 3 = shares a room but does not cosleep.

`sleep`

The parent’s estimate of the average hours of sleep the child receives in a 24-hour period.

Download `child-sleep-data.csv` and then complete the following steps:

2.3.1 Import and Inspect the Data

- Import `child-sleep-data.csv` into R; assign it a meaningful name.
- Inspect the `data.frame` using `str()`, `head()`, and `View()`.

2.3.2 Convert Categorical Variables to Factors

- Use `as.factor()` and `factor()` to convert any categorical variables to factors with the appropriate levels and labels.

2.3.3 Subsetting and Logical Expressions

Use `[]` to subset the data frame and find the answers to following questions/complete the tasks described below. You will also need to use `is.na()`, `max()`, `min()`, `mean()`, and logical expressions.

- What campus was the 48th participant recruited at?
- Return the data for participants who are missing responses on the sleep variable.
- What proportion of participants are missing data for the sleep variable?
- On average, how many hours do the children in this sample sleep per day?
- What is the average daily sleep of children in each of the three conditions?
- What is the participant ID of the participant(s) who reported the most sleep? The least sleep?

2.3.4 Participant Information

Use `table()` to identify how many, and what proportion of participants belong to each level of the three categorical variables.

Chapter 3

Data Visualization

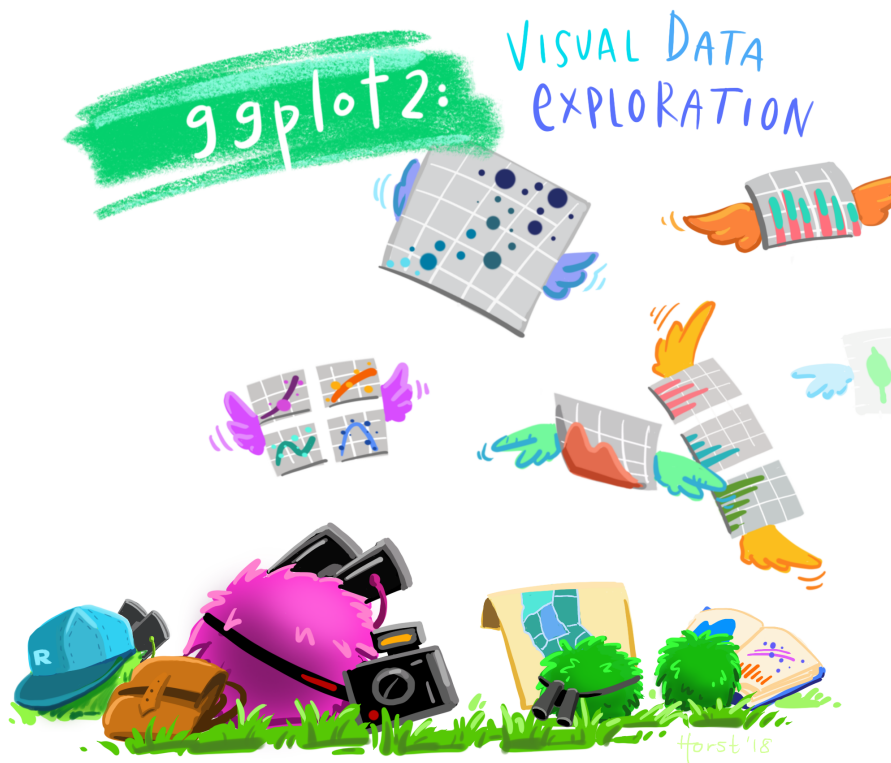


Figure 3.1: ggplot2

3.1 Learning Objectives

After completing this lab, you should be able to:

Install and load R packages.

Understand why data visualization is a useful tool for science communication.

Know and apply the APA guidelines for figures in APA manuscripts.

Understand the basic “grammar of graphics” used by `ggplot2`.

Apply the basic “grammar of graphics” to visualize data in R.

3.2 Prepare for the Lab

Ensure that you are comfortable with the content in labs 1 and 2.

3.3 Lab Activity

Use the `anscombe_long.csv` data for this lab activity.

3.3.1 Setup

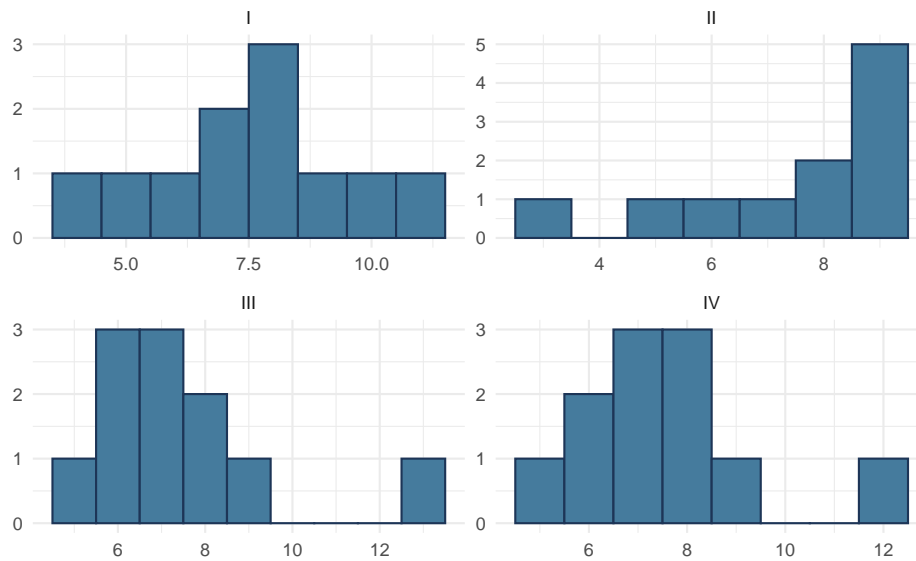
1. If you haven’t already, install `ggplot2`.
2. Load `ggplot2`.
3. Import `anscombe_long.csv` into R as a `data.frame` named `anscombe_long`.
4. Convert `anscombe_long$dataset` to a factor with levels: 1 = “I”, 2 = “II”, 3 = “III”, and 4 = “IV”.

3.3.2 Plots

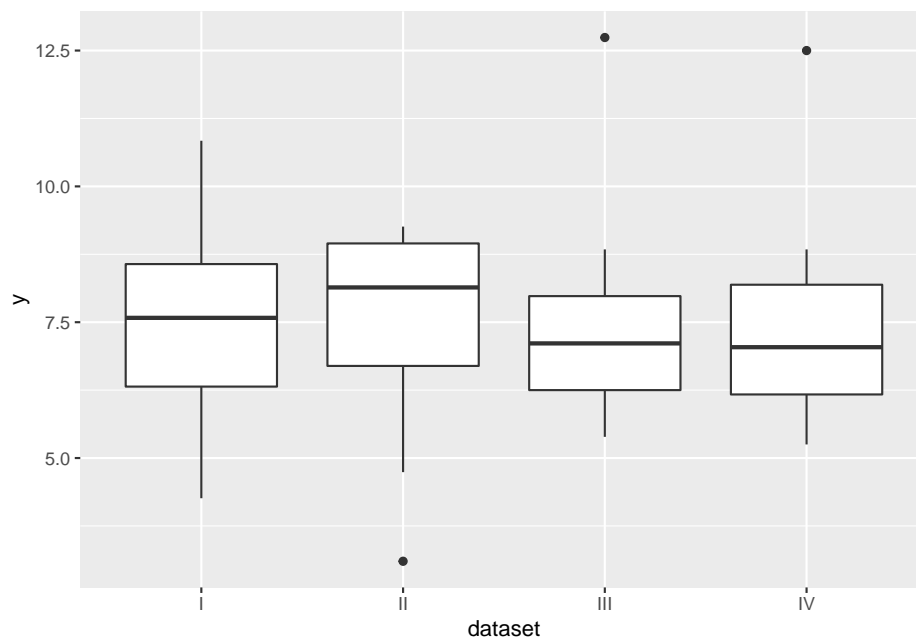
Write a script to produce each of the following plots. All of them use the dataset “`anscombe_long.csv`”.

3.3.3 Plot 1: Histograms

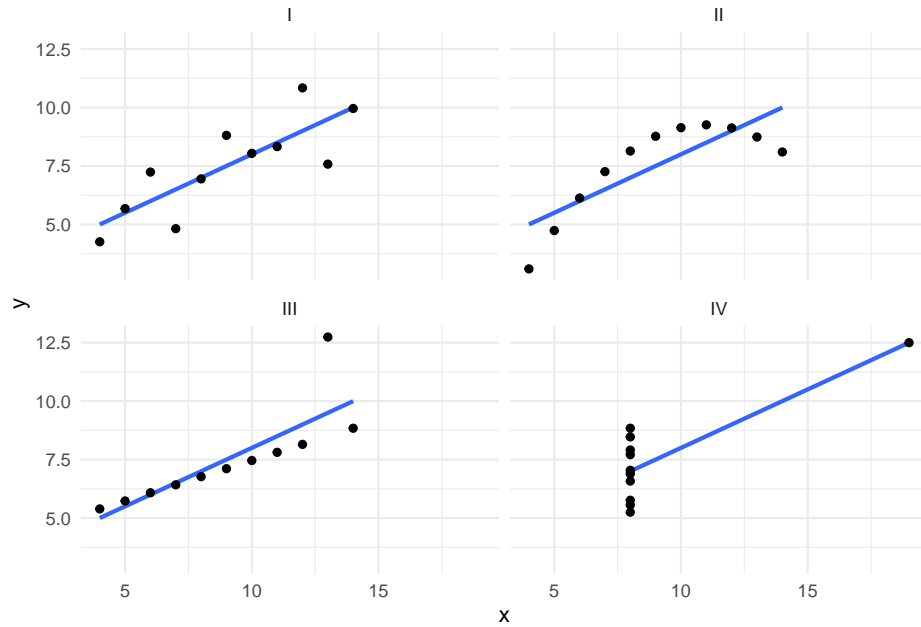
Anscombe's Quartet
Distributions of y



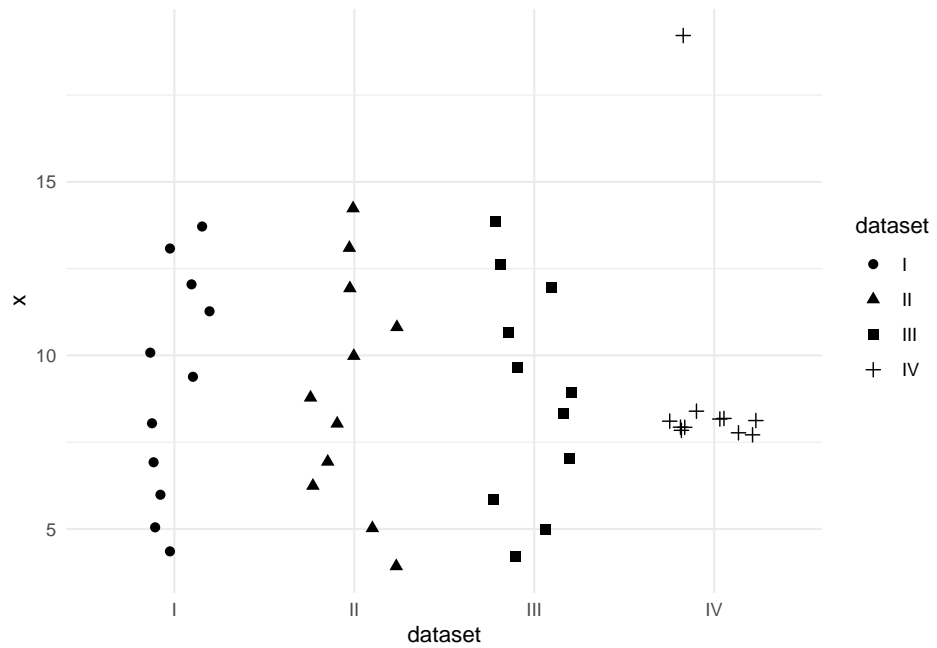
3.3.4 Plot 2: Boxplot



3.3.5 Plot 3: Scatter Plot



3.3.6 Plot 4: Jitter Plot



Tips for Recreating Plot 4

Identify the aesthetic mappings before you start.

- Which variable is mapped to x?
- Which is mapped to y?
- Are there any other aesthetic mappings?

Use `geom_jitter()` to plot the points. `geom_jitter()` is a variant of `geom_point()` that adds a little... jitter. It's useful when points would otherwise be overlapping. Note that the jitter is random, so your plot will not match this one exactly. The jitter can also affect the range of the scale axes.

Set the width of `geom_jitter()` to be equal to 0.25.

This plot uses `theme_minimal()`. You are free to use whatever theme you prefer.