Data Cleaning

Nicholas Ray

2024-09-23

Tests

When deciding on a hypothesis test, the "goodness" of a test is typically determined by evaluating the probability of rejecting the null hypothesis if it were true (denoted α and called the *level* or *size* of a test [type I error]) and the probability of failing to reject the null hypothesis if it were false (denoted β [type II error]).

There is a trade off between the probabilities α and β. If you decrease the level of significance such that α is smaller, it becomes harder to reject the null and you will fail to reject a truly false null hypothesis more often (i.e., higher β).

Typically, researchers predetermine the desired probability of rejecting a true null hypothesis (e.g., $\alpha = 0.05$) and try to minimize the probability of failing to reject a false null hypothesis- or equivalently- maximize *power* ($1 - \beta$, the probability of rejecting a false null hypothesis). That is, between two tests with identical α, we would prefer the test with highest power (smaller β). For a given α, power increases in sample size and the distance between the alternative and null hypotheses (i.e., all else equal, it is easier to distinguish between an alternative of 10 and null of 0 than between an alternative of 1 and a null of 0).

t-test

But we have already settled on our preferred test for right now (t-test, using $\frac{\overline{X}-\mu_0}{\hat{\sigma}[\overline{X}]}$). If we want to know how powerful this test would be for a given alternative ($\overline{X}$) and hypothesized null (μ_0), we need to fix α and know $\hat{\sigma}[\overline{X}]$. For $\overline{X} = 1$, $\mu_0 = 0$, $\alpha = 0.05$ and $\hat{\sigma}[\overline{X}] = 0.5$, our test statistic (t-stat or T) for our t-test would be:

$$\frac{\overline{X} - \mu_0}{\hat{\sigma}[\overline{X}]} = \frac{1 - 0}{0.5} = 2.$$

We know this t-stat is distributed standard normal, so α gives us known critical values (z). To figure out our theoretical power, or the probability of rejecting the null if were truly false, we need the probability that our t-stat falls within the rejection region given by z. For a two tailed test, our test statistic falls in the rejection region when $T < -z$ and $T > z$. So, we want to (sort of) plug in the areas $-T - z$ and $T + z$ into the cumulative distribution function (CDF) for the standard normal (Φ). Because the CDF is evaluated as $Pr[x \leq X]$, we need to evaluate the distance between T and z at 1-CDF to find something like $T + z$.

```
#t
t=(1-0)/0.5
#z
z=qnorm(0.05/2,lower.tail=F)
#pr(t<-z) (CDF evaluated at -T-z)
pnorm(-z-t)
```

```
## [1] 3.748053e-05
```

```
#pr(t>z) (1-CDF evaluated at t+z)
1-pnorm(z-t)
```

```
## [1] 0.5159678
```

```
#together:
pnorm(-z-t)+1-pnorm(z-t)
```

```
## [1] 0.5160053
```

For more details, please see Matt's solution to homework 2.3 when it is uploaded.

Cleaning Data

The rest of lab will be discussing basic cleaning skills, mostly in `tidyverse`. In the first section we will talk through chapter 3 from Wickham, Cetinkaya-Rundel, and Grolemund (2023), and I encourage you to read it because they do it better than me.

R for Data Science, Chapter 3

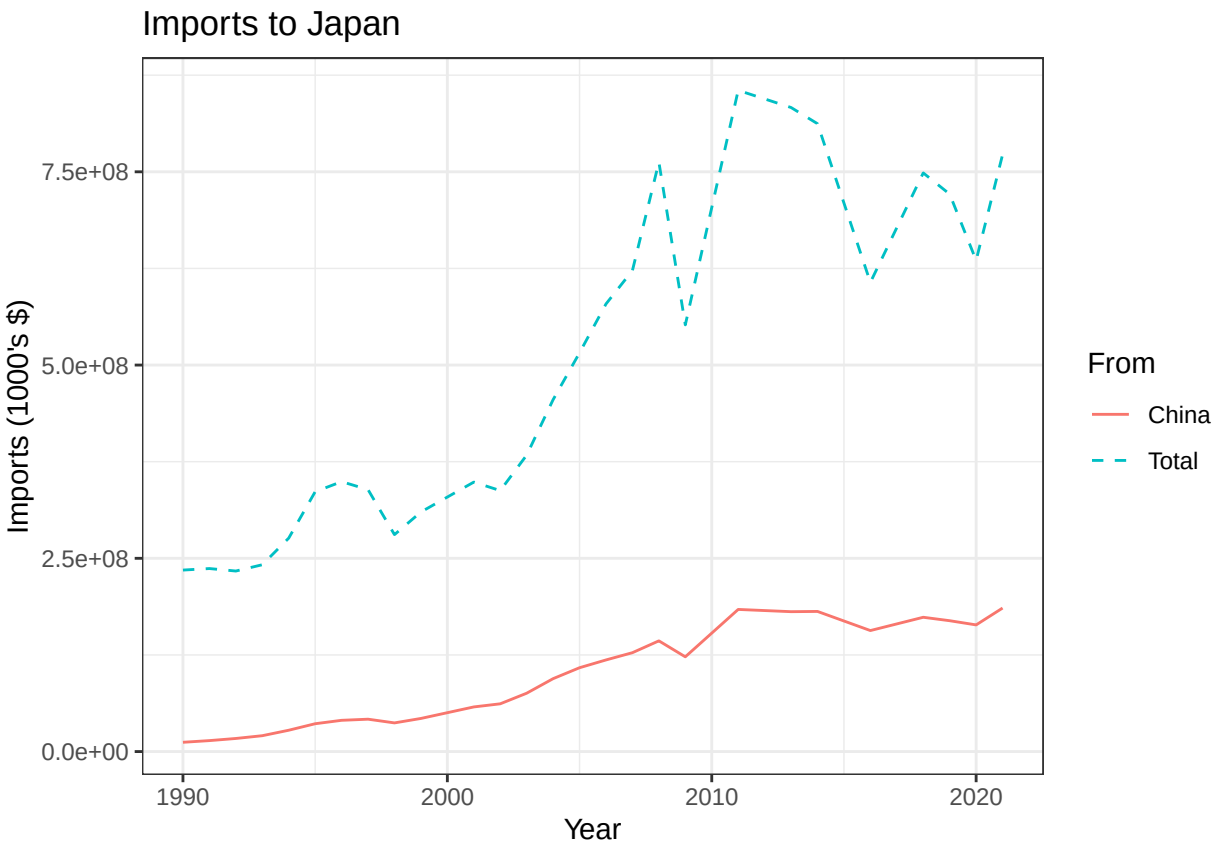
```
library(nycflights13)
library(tidyverse)

flights<-nycflights13::flights
View(flights)

# operations on rows ##############################################################
flights |>
  filter(carrier == "DL")
flights |>
  arrange(month, day, dep_time)
flights |>
  distinct(origin, dest)
flights |>
  count(month, day)

# operations on columns ###########################################################
flights |>
  mutate(speed = distance / air_time * 60)
flights |>
  select(arr_time)
flights |>
  rename(departure = dep_time)
flights |>
  relocate(day, month, year)

# multiple verbs, group_by() ######################################################
```

```r
flights |>
  filter(carrier == "DL") |>
  group_by(month) |>
  arrange(day) |>
  View()

# summarize() ##############################################################################
flights |>
  filter(carrier == "AA" | carrier == "DL") |>
  group_by(year, month, day, carrier) |>
  summarize(average_delay = mean(dep_delay, na.rm=T))
```

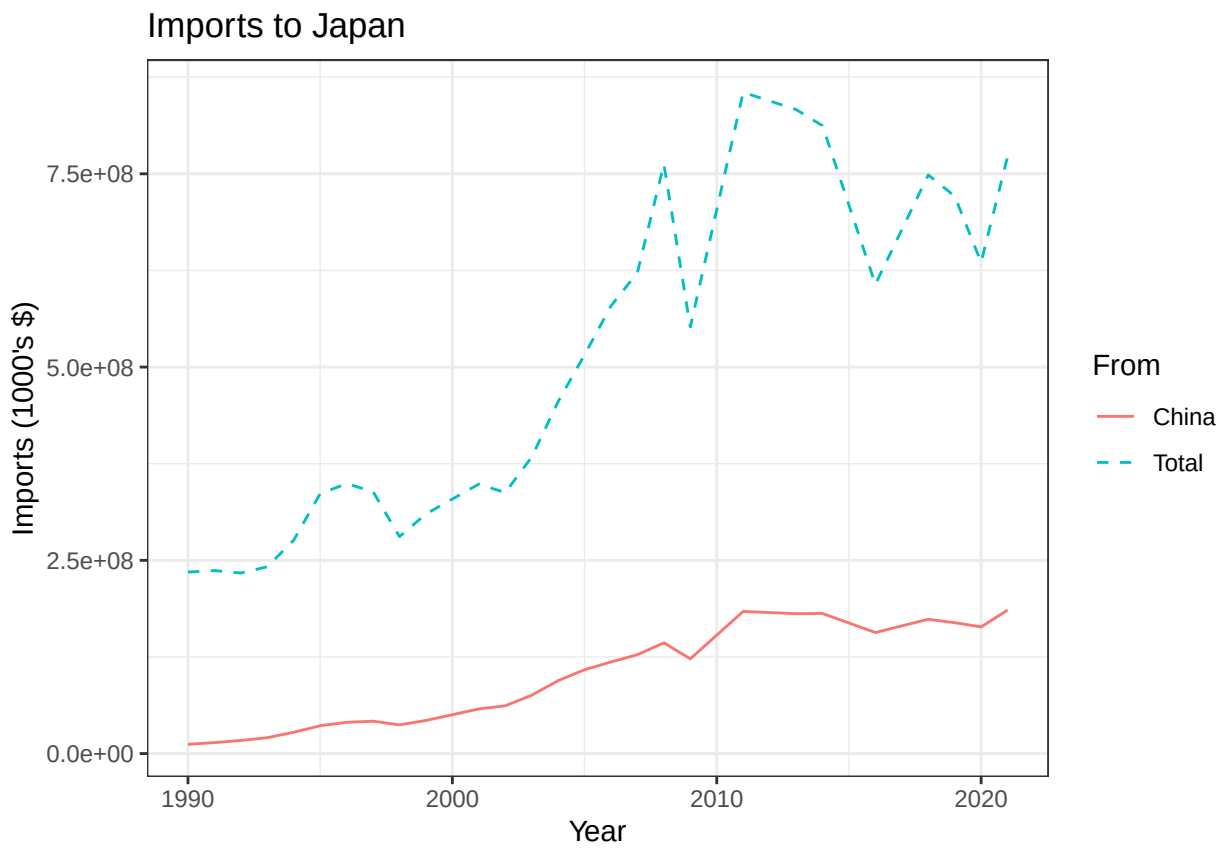
Another Example

Here's another example of cleaning somewhat "wild" trade data (OECD 2024). I typically do things somewhat differently (everyone does) and it may not be the best but I want to share some tricks with you.

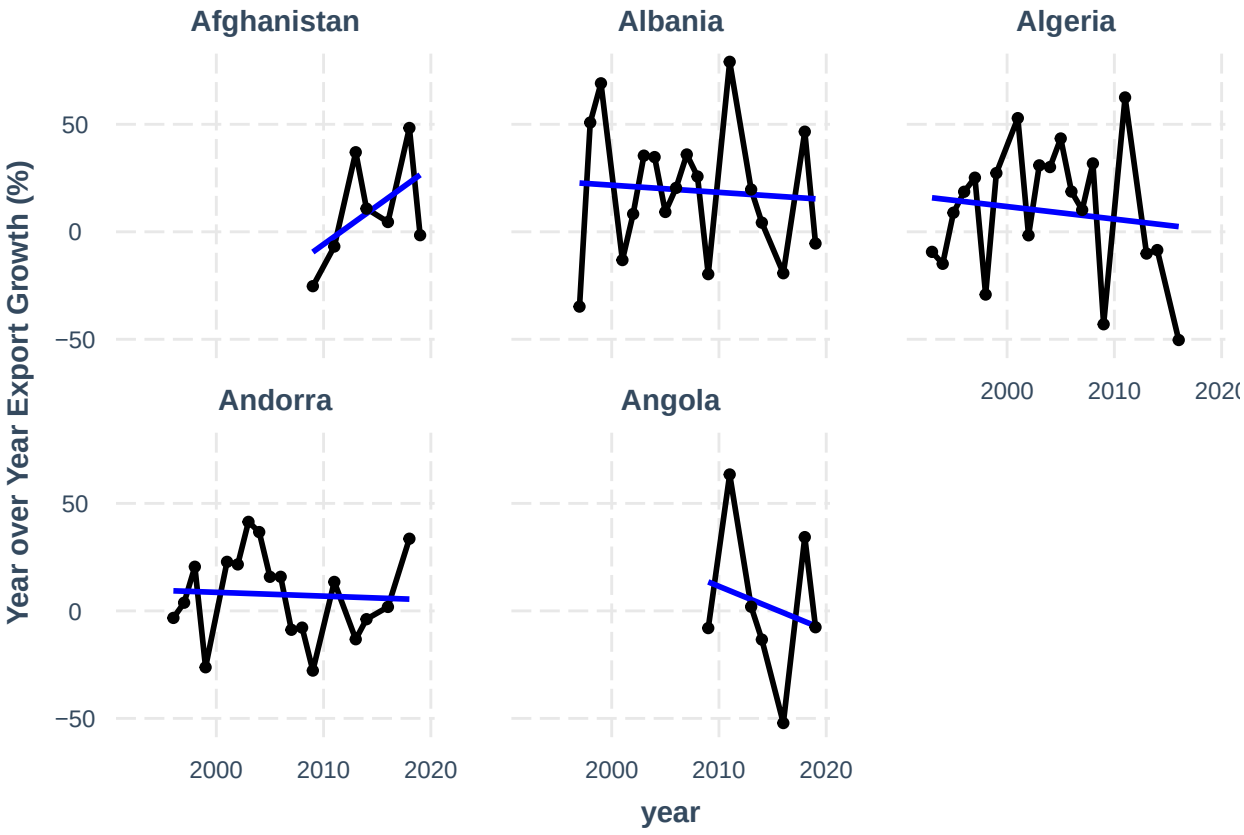
```r
#using the here() package
library(here);library(tidyverse)
trade<-read.csv(here("data","trade.csv"))

#useful base R way to edit rows that match a certain criteria
trade["Partner.country"][trade["Partner.country"]=="China (People's Republic of)"]<-"China"

#select() can do the same thing as rename() and relocate() at once, and use column numbers
trade<-trade %>%
  select(country=6,year=TIME_PERIOD,type=Flow,partner=10,value=19)

#case_when() is tidyverse's ifelse() and is very useful for creating new variables based only on some r
imports<-trade %>%
  filter(type=="Imports") %>%
  group_by(country,year) %>%
  mutate(imports_from_china=case_when(partner=="China"~value))

#a sometimes better combo of base R and tidyverse to do a similar thing
imports<-trade %>%
  filter(type=="Imports") %>%
  group_by(country,year) %>%
  mutate(imports_from_china=sum(value[partner=="China"],na.rm=T),
         total_imports=sum(value[partner=="World"],na.rm=T))

#can operate on a data.frame within a function
ggplot(filter(imports,country=="Japan")) +
  geom_line(aes(x=year,y=imports_from_china,color="China"))+
  geom_line(aes(x=year,y=total_imports,color="Total"),linetype="dashed")+
  labs(title="Imports to Japan",x="Year",y="Imports (1000's $)",color="From")+
  theme_bw()
```




```
#can also feed data.frame straight to ggplot()
imports %>%
  filter(country=="Japan") %>%
  ggplot(.) +
  geom_line(aes(x=year,y=imports_from_china,color="China"))+
  geom_line(aes(x=year,y=total_imports,color="Total"),linetype="dashed")+
  labs(title="Imports to Japan",x="Year",y="Imports (1000's $)",color="From")+
  theme_bw()
```




```
#lagging rows is very useful
exports<-trade %>%
  filter(type=="Exports",partner=="World") %>%
  group_by(country) %>%
  arrange(country,year) %>%
  mutate(value_year_before=dplyr::lag(value),
         growth=((value-value_year_before)/value_year_before)*100)

#one useful function for plotting observations over time on several units
library(panelr)
countries<-unique(exports$country)
exports %>%
  filter(country %in% countries[1:5]) %>%
  line_plot(.,growth,id="country",wave="year",overlay=F,add.mean=T,line.size=1)+
  labs(y="Year over Year Export Growth (%)")
```



Bibliography

OECD. 2024. "Bilateral Trade in Goods by Industry and End-Use (BTDIxE)." https://stats.oecd.org/Index.aspx?DataSetCode=BTDIXE_I4.

Wickham, Hadley, Mine Cetinkaya-Rundel, and Garrett Grolemund. 2023. *R for Data Science (2e)*. O'Reilly Media. https://r4ds.hadley.nz/.