Sample Answer Lab 02: Measurement Scales, Statements and Big Data

Handout date: Monday, September 09, 2019

Due date: Wednesday, September 18, 2019 at the beginning of the lecture as hardcopy

This lab counts 4 % toward your total grade

Task 1: Identify and justify the measurement levels of several statistical variables (1 point)

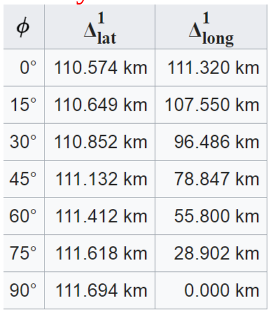
Justify your selection of the measurement scale. You may want to look up Wikipedia for some of the variables

a. The longitude and latitude in degrees on the earth's spherical surface. Be cautious in your arguments with regards to the longitude. That is, are distances between two longitudes constant all over the globe? (0.2 points)

Longitude: interval scaled only at a given latitude. Because longitudes and latitudes are circular measures, the selection of the origin is arbitrary. Therefore, they cannot be ratio scaled. Distances between two longitudes dependent on the latitude at which they are measured, using the cosine of the latitude as adjustment factor. The distance adjustment factor at the equator is $\cos(0°) = 1$ whereas at both pols it is $\cos(90°) = 0$. Distances between longitudes at a given latitude are comparable, however, distances between longitudes at different latitudes do not correspond with each other.

Latitude: Interval scaled. Because it is a circular measure, it does not have natural zero and the difference between two latitudes are almost identical with a slight elliptical distortion due the rotational forces exerted onto the earth. The table below show the distances between 1 degrees of latitude and longitude for an elliptical model (e.g., WGS84) of the earth surface. Therefore, the distances between 1° latitudes are not perfectly constant:

1.



ϕ	Δ^1_{lat}	Δ^1_{long}
0°	110.574 km	111.320 km
15°	110.649 km	107.550 km
30°	110.852 km	96.486 km
45°	111.132 km	78.847 km
60°	111.412 km	55.800 km
75°	111.618 km	28.902 km
90°	111.694 km	0.000 km

The $1/\cos(\text{latitude})$ adjusted longitudinal distances lead to the Mercator map projection.

b. The temperature on the Celisus scale. (0.1 points)

Interval scaled. The freezing and boiling points of water are defined as 0 and 100 degrees Celsius, respectively. It does not have natural zero so it is an interval variable like the temperature in Fahrenheit and Celsius.

c. The wind direction in degrees. (0.1 points)

Interval scaled. Because it is also a circular measure.

d. Number of break-ins in a neighborhood. (0.1 points)

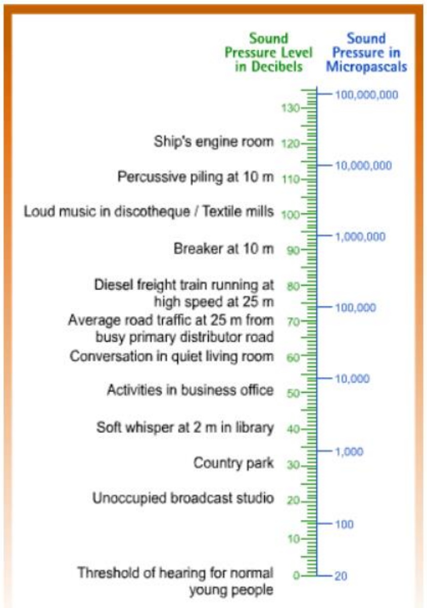
Ratio scaled. Discrete variable with a natural zero is meaningful and it make sense to calculate ratios.

e. The hierarchical classification of U.S. census enumeration units. (0.1 points)

Ordinal scaled. The smaller census enumeration units are embedded in larger units: Block < Block Groups < Tract < County < State < Region.

f. The decibel in acoustics (0.1 points)

Difficult to determine: ratio/interval. Decibel is used to measure sound pressure on a logarithmic scale (base 10). 0 dB does not mean no sound, it means the sound level being equal to be perceivable (the reference level on the logarithmic scale).



Relation between sound pressure in micropascals and sound pressure level in decibels re 20 μPa.

g. Income (0.1 points)

Ratio scaled. Income has natural zero with fixed intervals.

h. Income brackets for taxation purposes (0.1 points)

Ordinal scaled. The higher income class need to pay more tax, but the increments are not constant.

i. Elevation above sea-level at a fixed point in time. (0.1 points)

Ratio scaled. It has meaningful zero. However, over time this reference points shifts ranging from tides to climate associated sea-level changes.

Task 2: Working with Data (2 points)

For all tasks below show your properly formatted code. You find the necessary code for the examples in Lander. Only if asked show also the output.

a. Import the SPSS data-file `Concord1.sav` in the Lab02 folder as ***data-frame*** into the R environment by using a function from the library `foreign`. Make sure to name your data-frame properly. (0.2 points)

```
library(foreign)
concord<- read.spss(".//lab2//Concord1.sav", to.data.frame=TRUE)
```

b. Show the ***tail*** of the last 6 records of the imported data-frame. (0.1 points)

```
tail(concord, n=6)
```

add ouput

c. Show the ***summary*** information for the imported data-frame. (0.1 points)

```
summary(concord)
```

add summary output

d. Discuss the summary: How did the ***average*** water consumption changes from 1979 to 1981? (0.1 points)

Average water consumption reduces every year from 2,974 to 2,298 in 1979 to 1981.

e. Discuss the summary: Which variable has ***missing*** observations? (0.1 points)

The summary statistics indicate that `water79` variable has 47 missing observations. `retire` is a factor and the remaining variables are metric

f. Discuss the summary: Which variable is a factor? (0.1 points)

```
class(concord$retire)
[1] "factor"
```

Variable retire is a factor variable.

g. List all ***case numbers***, which have at least for one variable missing data. Show also the code. (0.2 points)

```
concord[is.na(concord$water79), ]$case
23  40  46 108 142 143 144 145 146 153 159 178 181 197 199 205
213 283 290 310 334 359 375 385 408 421 466 480 481 487 488 490
491 497 498 499 500 502 506 507 508 511 512 513 514 515 516
```

h. Which ***class*** are the following data selections: [a] **`Concord$retire`**, [b] **`Concord["retire"]`**, [c] **`Concord[ , "retire"]`**, and [d] **`Concord[["retire"]]`**? Show code and the output. (0.2 points)

Use the class function to determine the type of each statement:

[a]: factor,

[b]: data.frame,

[c]: factor,

[d]: factor

i. Calculate the ***average*** of the household's water consumption for the 3 years and save it the new variable **`meanWater`** into the data-frame. (0.2 points)

```
meanWater <- rowMeans(concord[c("water79","water80","water81")],
na.rm=T)

meanWater <- ifelse(is.na(concord$water79),
(concord$water80+concord$water81)/2,
(concord$water79+concord$water80+concord$water81)/3)
```

j. Remove the variable **`case`** from the data-frame by assigning it **`<- NULL`**. Show the code (0.1 point)

```
concord$case <- NULL
```

k. Add a new **`caseID`** variable by labeling each record by its record number ranging from 1 to the number of observations. Show the code. (0.1 point

```
concord$caseID <- 1: nrow(concord)
```

l. ***Bind*** the two variables **`peop80`** and **`peop81`** together into a ***matrix***. Show the code. (0.1 points)

```
waterCon <- cbind(concord$water79, concord$water80,
concord$water91)
```

m. Give a code example of the use of the **`ifelse`** statement. (0.1 points)

`ifelse` function: `ifelse(test, yes, no)`, essentially if the test is correct then do yes, otherwise do no.

Example:

```
ifelse(Concord$water81 <= Concord$water79,
"consumption decreases", "consumption increase")
```

n. Give an example of the **`while`** statement (0.1 points)

While loops are used to loop until a specific condition is met.

Example:

```
i<-1
while(i<10){
  print(Concord[i,])
  i <- i+1
}
```

o. What are [a] positional, [b] named and [c] default arguments of a function? (0.3 points)

See the user function `pow( )`, which powers a base by an exponent and optional with an inverse exponent:

```
pow <- function(base, expo, inv=FALSE){
  if (inv==FALSE) result <- base^expo else
    result <- base^(1/expo)
  return(result)
} # end::pow
```

[a] A positional argument of a function matches arguments by their positions, it is the most common and simplest one.

Example: `pow(8, 2)` :"8 raised to the power 2 is 64"

`pow(2, 8)` "2 raised to the power 8 is 256"

[b] A named argument of a function matches arguments by their names.

Example: `pow(base=8, expo=2)` :"8 raised to the power 2 is 64"

`pow(expo=2, base=8)` "8 raised to the power 2 is 64"

[c] Previous statements assumed the default argument inv=FALSE of the pow( ) function. Overwriting the default argument with inv=TRUE calculates the inverse power.

Example: `pow(4,2)`: squares the base 4

`pow(4,2, inv=TRUE)`: takes the square-root of 4

Task 3: Critical discussion of big data analyses (1 point)

Read the document **BIGDATAANDSTATITICS.PDF** in the **LECTURE01** folder.

[a] Summarize in your own words in a few sentences an example where and why big data analysis failed. (0.5 points)

Comment: In Google's flu research, engineers care about correlation more than causation so they do not detect what causes the phenomenon. The reason of the election example is that samples did not reflect the true population so sampling error and sampling bias become the problems. The last reason is the multiple-comparison problem that there are vastly more possible comparisons than there are data points to compare.

[b] In a few sentences provide some arguments why theory and statistics are still important in the era of big data analysis. (0.5 points)

Comment: If we ignore theory and statistics, both small data analysis and big data analysis have a high potential to fail, because both analyses will suffer from the same shortcomings. Due to the vast number of data, the potential statistical problems may even be exaggerated in big data analysis. In particular the multiple testing problem needs to be dealt with in big data analysis to judge, which results are truly significant. The discipline of statistics has evolved over time to deal with these problems or, at least, to acknowledge their potential dangers. Centuries of accumulated statistical knowledge should not be ignored by ad hoc big data fishing expeditions.

For the theory-free exploratory analyses of associations, researchers have no idea about what may have caused these observed associations. Therefore, the results may point to **spurious associations**. It is more important to understand, at least with some degree of confidence, as well as **test for** what is "known" about the **underlying data generating process** than just look for potentially spurious associations. Ultimately, big data analysis is not able to identify **cause and effect relationships**.