Data Analysis

Fabrice Niessen

2013-01-24 Thu

Contents

1. What is data?

Population set of items

Variable measurement or characteristic

Qualitative can be defined by a label and have discrete values

Quantitative measured on a numerical scale

Processed (**tidy**) data

- Each variable forms a column
- Each observation forms a row
- Each table / file stores data about one kind of observation

2. Representing data

Distribution = density? possible values of X and the probabilities for each value

- Defined by a set of parameters (sometimes represented by Greek letters)

Probability chance something will happen (Pr)

E[X] Mean (expected value, "center" of a distribution)

Var[X] Variance (how "spread out" a distribution is, in units of X2̂)

SD[X] Standard deviation (also how "spread out" a distribution is, in units of X) = root(Var[X])

Condition: fixed value for some parameter (X|u=2)

Distributions:

- **Binomial** bin(n = 10, p = 0.5): describes the sum of a set of coin flips
- **Normal** N(mu = mean, sigma = sd): probabilities for ranges of variables
- **Uniform** U(alpha = min, beta = max): all values (between boundaries) are equally likely

3. Simulation basics

- Important simulation functions

beta

binom

cauchy

chisq

exp

f

gamma

geom

hyper

logis

lnorm

nbinom

norm

pois

t

unif

weibull

4. Types of data analysis questions

In approximate order of difficulty:

1. descriptive :: describe a set of data
2. exploratory :: find relationships you didn't know about
3. inferential :: use a relatively small sample of data to say something about a bigger population
4. predictive :: use the data on some objects to predict values for another object
5. causal :: find out what happens to one variable when you make another variable change
6. mechanistic :: understand the exact changes in variables that lead to changes in other variables for individual objects

5. Structure of a Data Analysis

Steps:

1. Define the question
   - Start with a general question
   - Make it concrete (can I use quantitative characteristics?)
2. Define the ideal data set, depending on your goal

descriptive	a whole population
exploratory	a random sample with many variables measured
inferential	the right population, randomly sampled
predictive	a training and test data set from the same population
causal	data from a randomized study
mechanistic	data about all components of the system

3. Determine what data you can access
   - Find it
   - Buy it
   - Generate it
4. Obtain the raw data
5. Clean the data
   - If it is pre-processed, make sure you understand how

XXX Missing summary XXX

6. Organizing a Data Analysis

- Data
  - Raw data
  - Processed data
    - Shoud be named so it is easy to see which script generated the data
    - Should be **tidy**
- Figures
  - Exploratory figures, not necessarily part of your final report
  - Final figures (clear axes/colors)
- R code
  - Raw scripts
  - Final scripts (clearly commented, including processing details)
  - R Markdown files (optional)
    - Can be used to generate reproducible reports
    - Text and R code are integrated
- Text
  - Readme files (should contain step-by-step instructions for analysis, not necessary if you use R Markdown)
  - Text of analysis should tell a story (don't report analysis in chronological order: Analyses should be reported in an order to convey the story being told with the data analysis)
    - Title
    - Introduction (motivation)
    - Methods (statistics you used)
    - Results (including measures of uncertainty)
    - Conclusions (including potential problems)
- Further resources
  - Reproducible research and Biostatistics
  - Managing a statistical analysis project guidelines and best practices
  - Project template - a pre-organized set of files for data analysis

7. Getting Data

- Get/set your working directory
- Types of files
  - Tab-delimited
  - CSV
  - Excel
  - JSON
  - HTML/XML
  - Database

download.file() download a file from the Internet Be sure to record when you downloaded

```
fileUrl <- "http://..."
download.file(fileUrl,destfile="./data/file.csv",method="curl")
```

read.table() read data into R

```
data <- read.table("./data/file.txt",sep="\t",header=T,quote="")
```

- read.csv()

  read.csv(file.choose()) pick a file

  read.csv2()

```
data <- read.csv("./data/file.csv")
```

```
con <- file("./data/cameras.csv", "r")
cameraData <- read.csv(con)
close(con)
```

read.xlsx(), read.xlsx2() read Excel files

readLines() read lines of text from a connection (**remember to close connections**)

```
con <- url("http://scholar.google.com/citations")
htmlCode <- readLines(con)
close(con)

# get data off webpages
html3 <- htmlTreeParse("...")
xpathSApply(html3, "//title", xmlValue) # find value of title tag
xpathSApply(html3, "//td[@id='col-citedby']", xmlValue) # access parts of the tab
```

write.table() write data

- Further resources: packages

  httr for working with HTTP connections

  RMySQL for interfacing with mySQL

  bigmemory for handling data larger than RAM

  foreign for getting data into R from SAS, SPSS, Octave, etc.

8. Data Resources

List of cities/states with open data

9. Summarizing data

- Look for
  - Missing values
  - Values outside of expected ranges
  - Values that seem to be in the wrong units
  - Mislabeled variables/columns
  - Variables that are the wrong class
- Looking at data

```
dim(data)
names(data)
nrow(data)
ncol(data)
```

```
quantile(data,na.rm=TRUE)
summary(data)
```

```
class(data)
sapply(data[1,], class) # class of each individual column (tricky way)
```

```
unique(data$var)
length(unique(data$var)) # how many unique values
table(data$var) # number of times each unique value appears
```

```
table(data$var1,data$var2) # look at the relationship between var1 and var2
```

```
any(data$var > 40) # check if any value is greater than 40
all(data$var > 40) # check if all value are greater than 40
```

```
data[data$var1 > 0 & data$var2 > 0, c(1,2)] # subset on var1 and var2
data[data$var1 > 0 | data$var2 > 0, c(1,2)] # subset on var1 or var2
```

```
is.na(data$var) # determine if there are missing values
sum(is.na(data$var)) # calculate the number of NA values (sum of TRUE)
table(is.na(data$var)) # how many NA values and how many non-NA values
table(data$var,useNA="ifany") # show missing values as well
```

```
colSums(data) # sum of each column
colMeans(data,na.rm=TRUE) # means of each column (ignore NA values)
rowSums(data,na.rm=TRUE)
rowMeans(data,na.rm=TRUE)
```

10. Data munging basics (= key process)

- (Partial list of) Munging operations
  - **These steps must be recorded** in their own R script

- **90% of your effort will often be spent here**

- Fix variable (data frame) names

```
tolower(names(data))
```

```
splitNames <- strsplit(names(data),"\\.") # split at period in names
firstElement <- function(x){x[1]} # select the 1st component of the vector
sapply(splitNames,firstElement) # remove everything after the trailing dot
```

```
gsub("_","",names(data))
```

- Create new variables

```
data$ranges <- ranges # add a variable to the data frame
```

- Merge data sets

```
mergedData <- merge(dfx,dfy,by.x="idx",by.y="idy",all=TRUE)
```

- Reshape data sets

Replace **quantitative variables** with qualitative variables that are labelled with the **ranges** where they appear because:

- you want to look at variables at a glance
- you think you don't necessarily believe the precision of the quantitative variables

```
ranges <- cut(data$var,seq(0,3600,by=600))
table(ranges,useNA="ifany")
```

Break the values up by quantile

```
ranges <- cut2(data$var,g=6) # 6 ranges of approximately an equal number of point
table(ranges,useNA="ifany")
```

Sort values

```
sort(data$var)
```

Order values

```
order(data$var) # says how you need to rearrange the vector
data$var[order(data$var)] # take the indices of the order command
```

Reorder a data frame

```
sortedData <- data[order(data$var),]
```

Reorder by multiple variables

```
sortedData <- data[order(data$var1,data$var2),]
```

Reshape data

```
melt(misShape,id.vars="",variable.name="",value.name="")
```

- Deal with missing data
- Take transforms of data
- Check on and remove inconsistent values

11. Exploratory Graphs

- Why do we use graphs in data analysis?
  - To understand data properties
  - To find patterns in data
  - To suggest modeling strategies
  - To "debug" analyses
  - To communicate results (to other people)
- Characteristics of exploratory graphs
  - They are made quickly
  - A large number are made
- Use position comparison on a common scale See Graphical Perception and Graphical Methods for Analyzing Scientific Data[1]
- Boxplot

```
boxplot(data$var,col="blue")
boxplot(data$x ~ as.factor(data$y),col="blue") # x broken down by level y
boxplot(data$x ~ as.factor(data$y),varwidth=TRUE) # width proportional to # obs
```

- Barplot

```
barplot(table(data$var),col="blue") # barplot of the levels (number of observatio
```

- Histogram (**distribution** of data, **frequency**)

  Frequency = exact number

```
hist(data$var,col="blue") # shape of distribution
hist(data$var,col="blue",breaks=100) # much more fine-grained distribution
```

  Report where the distribution is centered:

```
meanValue <- mean(data$var)
lines(rep(meanValue,100),seq(0,<i>,length=100),col="red",lwd=5)
```

- Density plot (smoother histogram)

  Density = percentage of observations

```
dens <- density(data$var)
plot(dens,lwd=3,col="blue")
```

  Easy to compare multiple distributions:

```
# add another density
densMales <- density(data$var[which(data$sex==1)])
lines(densMales,lwd=3,col="orange")
```

[1]http://webspace.ship.edu/pgmarr/Geo441/Readings/Cleveland and McGill 1985 - Graphical Perception and Graphical Methods for Analyzing Scientific Data.pdf

- Scatterplot (to visualize relationships between variables – for exploratory data analysis)

  ```
  plot(data$x,data$y,pch=19,col="blue")
  ```

  You can pick out particular problems with the data or weird patterns in the data.

  ```
  plot(data$x,data$y,pch=19,col="blue",cex=0.5) # smaller points
  ```

  Use color to **expand beyond 2 dimensions** – See how the relationship between 2 variables is different based on a **3{̂rd} variable**:

  ```
  plot(data$x,data$y,pch=19,col=data$sex,cex=0.5) # col param must be a vector of i
  ```

  or

  ```
  plot(data$x,data$y,pch=19,col=((data$sex=="Male")*1+1)) # male in black, female i
  ```

  Use size is useful:

  ```
  percentMaxAge <- data$age/max(data$age)
  plot(data$x,data$y,pch=19,col="blue",cex=percentMaxAge*0.5) # size of points prop
  ```

  But it's easier to see if you **break numeric variables as factors**:

  ```
  library(Hmisc)
  ageGroups <- cut2(data$age,g=5)
  plot(data$x,data$age,pch=19,col=ageGroups,cex=0.5) # see if the relation is modif
  ```

  Overlay lines/points:

  ```
  lines()
  points()
  ```

  If you have a lot of points:

  ```
  plot(x,y,pch=19)
  ```

  - Sample the values (take a random subset)

    ```
    sampledValues <- sample(1:1e5,size=1000,replace=FALSE)
    plot(x[sampleValues],y[sampledValues],pch=19)
    ```

  - Use a smoothScatter (smooth density plot)

    ```
    smoothScatter(x,y,pch=19)
    ```

  - hexbin package

    ```
    library(hexbin)
    hbo <- hexbin(x,y)
    plot(hbo)
    ```

- QQ-plot

  ```
  qqplot(x,y) # plot the quantile of x vs the quantile of y
  abline(c(0,1)) # intercept and slope (here: 45-degree line)
  ```

- Matplot (spaghetti-plot)

  If you **observe data over the time**, compare trends or trajectories over time by looking at the trajectories across columns:

```
matplot(matrix,type="b") # plot each column as one specific line
```

- Heatmap (= sort of 2-D histogram)

  Color represents intensity (the brighter, the larger value):

  ```
  image(1:10,161:236,as.matrix(data[1:10,161:236]))
  ```

  A little bit confusing: it sort of transposes the data in the image. To match a little bit more your intuition of the matrix (rows and columns), take the **transpose** of it:

  ```
  newMatrix <- as.matrix(data[1:10,161:236])
  newMatrix <- t(newMatrix)[,nrow(newMatrix):1]
  image(161:236,1:10,newMatrix)
  ```

- Geographic maps

  ```
  library(maps)
  map("world")
  points(lat,lon,col="blue",pch=19)
  ```

- Missing values and plots

  Keep in mind that **R skips over NA values**: it's not plotting those points, they don't appear on the plot.

  Deal with that fact if you don't want to get inappropriate or incorrect conclusions.

  Explore the data and see if there are relationships between missing values and the other variables in your data set:

  ```
  x <- rnorm(100)
  y <- rnorm(100)
  y[x < 0] <- NA
  boxplot(x ~ is.na(y))
  ```

12. Expository graphs

- Graphs that you're likely to show, **to communicate results**
- Axes

  ```
  plot(x,y,pch=19,col="blue",xlab="Label (unit)",ylab="Label (unit)")
  plot(x,y,xlab="Label (unit)",ylab="Label (unit)",cex.lab=2) # labels legends
  plot(x,y,xlab="Label (unit)",ylab="Label (unit)",cex.axis=1.5) # axis legends
  ```

  Add a legend:

  ```
  legend(xpos,ypos,legend="Text",pch=19,cex=0.5)
  legend(xpos,ypos,legend=c("men","women"),col=c("black","red"),pch=c(19,19),cex=c(
  ```

  or (put the legend in a second plot)

  ```
  par(mfrow=c(1,2))
  plot(d$x,d$y,col=as.numeric(d$factor),pch=19)
  plot(1:10,type="n",xaxt="n",yaxt="n",xlab="",ylab="")
  legend(1,10,legend=unique(d$factor),col=unique(as.numeric(d$factor)),pch=19)
  ```

  Add a title:

```
plot(x,y,main="Type of plot or Conclusion of the plot")
```

Add text to multiple panels:

```
par(mfrow=c(1,2))
...
mtext(text="(a)")
...
mtext(text="(b)")
```

Figure captions (bolded text describes the whole purpose of the entire plot).

Check for colorblindeness: Vischeck

- Devices

  pdf(file="file.pdf") create a PDF file

  png(file="file.png") create a PNG file

  dev.copy2pdf save the file you see on the screen

- Something to avoid: the top ten worst graphs

13. Hierarchical clustering

- Clustering
  - Organize things that are **close** (not very well distinguished) into groups
  - Find the closest two observations together, put them together, find the next two
- How do we defined close? Pick a distance/similarity that makes sense for your problem
  - Continuous – Euclidian distance
  - Continuous – Correlation similarity
  - Binary – Manhattan distance (shortest distance between 2 blocks)

dist(df) calculate the distances between the observations (points)

```
dataFrame <- data.frame(x=x,y=y)
distxy <- dist(dataFrame)
hClustering <- hclust(distxy)
plot(hClustering)
```

Prettier dendograms (where you can color the leaves): `myplclust()`

```
dataFrame <- data.frame(x=x,y=y)
distanceMatrix <- dist(dataFrame)
hClustering <- hclust(disanceMatrix)
myplclust(hClustering,lab.col=numVar) # numeric factor
```

See colored dendogram at http://gallery.r-enthusiasts.com/.

- Heatmap

```
dataMatrix <- as.matrix(dataFrame)
heatmap(dataMatrix)
```

14. K-means clustering

- Partitioning approach
  - Divide all of the points into a fixed number of clusters (must be known in advance)
  - Guess where the centers might be

```
kClust <- kmeans(dataFrame, centers=3, nstart=100)
# average on 100 random starts, so that points gets classified much more
# stably to a specific cluster (cluster name are not necessarily stable)

names(kClust)
kClust$cluster
```

```
table(kClust$cluster,dataFrame)
```

```
plot(x,y,col=kClust$cluster)
points(kClust$centers,col=1:3)
```

```
plot(kClust$center[<i>],pch=19,ylab="Cluster Center",xlab="")
```

15. Dimension reduction

- 2 popular dimension reduction techniques (to compress a large set of variables and compress them down into a smaller set of variables that are easier to interpret)
  - Principal components analysis
  - Singular Value decomposition
- Add a particular pattern

Too complex for me!

16. Clustering example

```
numVar <- as.numeric(as.factor(df$var))
```

17. Least Squares

- Goals of statistical modeling (regression analysis?)
  - Describe the distribution of variables
  - Describe the relationship between variables
  - Make inferences about ditrisbutions or relationships
- **Jittered plot**: add a tiny little bit of random noise to the variables, so that the plotted points are not stacked anymore on each other

```
plot(jitter(d$x,factor=2),jitter(d$y,factor=2),pch=19, col="blue")
```

- Fit a line to the data

  Sum((Ci - mu)2̂) must be minimized

  ```
  plot(d$x,d$y,pch=19,col="blue")
  lm1 <- lm(d$y ~ d$x) # least squares (quantitative outcome ~ covariant variables)
  lines(d$x,lm1$fitted,col="red",lwd=3)
  # or abline(lm1,col="red",lwd=3)
  ```

  Line: y_i = b_0 (**intercept**) + b_1 (**slope**) x_i

  A **one** unit increase in x is associated with a <slope> unit increase in y.

  ```
  lm1
  # intercept = lm1$coeff[1]
  # slope = lm1$coeff[2]
  ```

- Allow for variation

  Line: y_i = b_0 + b_1 x_i + e_i where e_i is everything we didn't measure (error term)

  `lm` is solving this!

- Plot **what is leftover** (after you've substracted the regression line out of the points = **residuals**)

  ```
  plot(d$x,lm1$residuals,pch=19,col="blue")
  # differences between the points and the line
  abline(c(0,0),col="red",lwd=3)
  ```

18. Inference

- Create a "population" of 1 million rows
- Histogram of estimates for intercept and slope
- Central limit theorem

  ```
  lm1 <- lm(d$y ~ d$x) # estimate intercept term, then estimate slope for x
  summary(lm1)
  ```

- Degrees of freedom ≚ number of samples - number of things you estimated

```
lines(x,dt(x,df=3),lwd=3,col="red") # t-distribution
lines(x,dt(x,df=10),lwd=3,col="red") # 10 degrees of freedom
```

- **Confidence intervals**: something about how good our estimate is

  ```
  confint(lm1,level=0.95)
  ```

  95% CI: <value at 2.5%> unit - <value at 97.5%> unit.

19. P-values

- P-values
  - Most common measure of "statistical significance"

    - Used for **decision making**
    - Controversial among statisticians

- Defined the hypothetical distribution of a data summary when “nothing is going on” = **Null distribution** (slope = 0)
- Calculate the summary with the summary we have (**observed statistic**)
- Compare what we calculated to our hypothetical distribution and see if the value is “extreme” (p-value)
- The difference is:

  P < 0.05 significant

  P < 0.01 strongly significant

  P < 0.001 very significant

- **Be careful!** P > 0.05 for each case, but once out of 20 times, it will be < 0.05 even if nothing is going on (false positive)...

20. Regression with factor variables

- Fitting lines = fitting means (in factor modeling approach)
- Average score by rating

```
plot(movies$score ~ jitter(as.numeric(movies$rating)),col=blue,xaxt="n",pch=19)
axis(side=1,at=unique(as.numeric(movies$rating)),labels=unique(movies$rating)
meanRatings <- tapply(movies$score,movies$rating,mean)
points(1:4,meanRatings,col="red",pch="-",cex=5)
```

```
lm1 <- lm(movies$score ~ as.factor(movies$rating))
summary(lm1)
```

- Plot fitted vlaues

anova analysis of variance table

- Tukey’s Honestly Significant Difference test

21. Multiple variable regression

- Color by male/female

```
lmM <- lm(d$y[d$Sex=="Male"] ~ d$x[d$Sex=="Male"])
lmF <- lm(d$y[d$Sex=="Female"] ~ d$x[d$Sex=="Female"])
plot(d$x,d$y,pch=19)
points(data$x,data$y,pch=19,col=((data$sex=="Male")*1+1))
lines(d$x[d$Sex=="Male"],lmM$fitted,col="black",lwd=3)
lines(d$x[d$Sex=="Female"],lmM$fitted,col="red",lwd=3)
```

22. Regression in the real world

- Things to pay attention

- **Confounders** = variable that is correlated with both the outcome and the covariates
- **Complicated interactions**
  - X correlated with Y

    ```
    anova(lm(d$y ~d$x))
    ```

  - Strong correlation if very large F value and very tiny P value
- **Skewness**

  Logs to address right-skew dat

  ```
  hist(log(data+1),col="blue",breaks=100) # + 1 if data = 0
  ```

- **Outliers** = data points that do not appear to follow the pattern of the other data points. . . but they **may be real**!

  If you know they aren't real/of interest, remove them (but changes question!)

  Alternatively:

  - Sensitivity analysis – Is it a big differnece if you leave it in/take it out?
  - Logs – if the data are right skewed (lots of outliers)
  - Robust methods – More robust approaches: `Robust`, `rlm`
- **Non-linear patterns**: A line isn't always the best summary

  http://en.wikipedia.org/wiki/Linear_regression
- **Variance changes**
  - Box-Cox transform
  - Variance stabilizing transform
  - Weighted least squares
  - Huber-white standard errors
- **Units/scale isssues**: makes more sense to standardize in relative units (number of death per million inhabitants) than in absolute units (number of deaths)
- **Overloading regression**
- **Correlation and causation**: be critical of surprising associations, consider alternative explanations