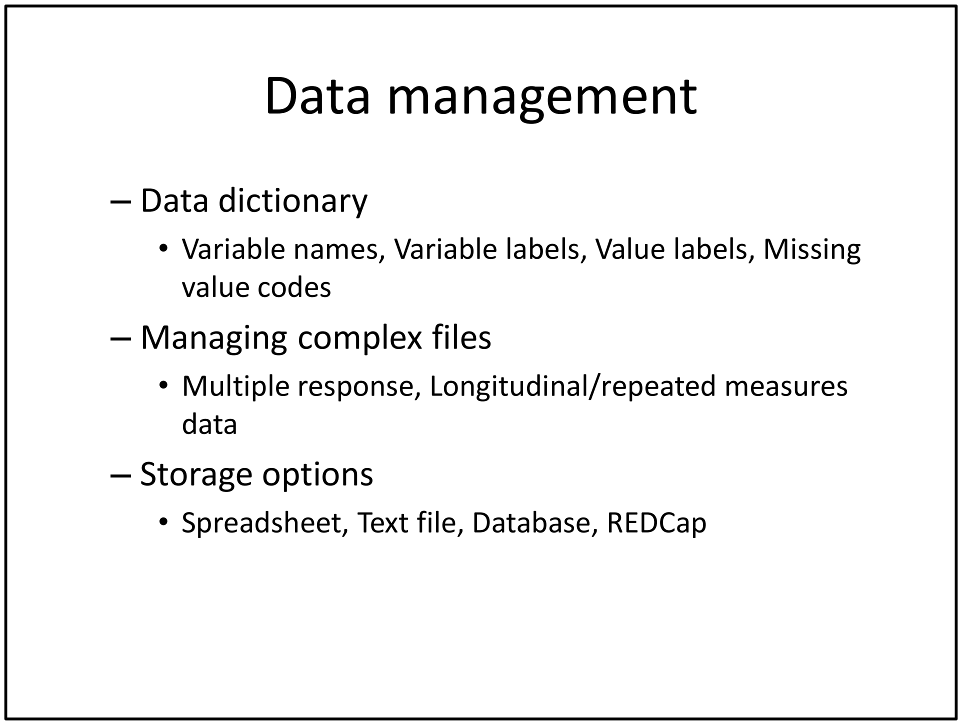
Video 11 - Data management

Steve Simon

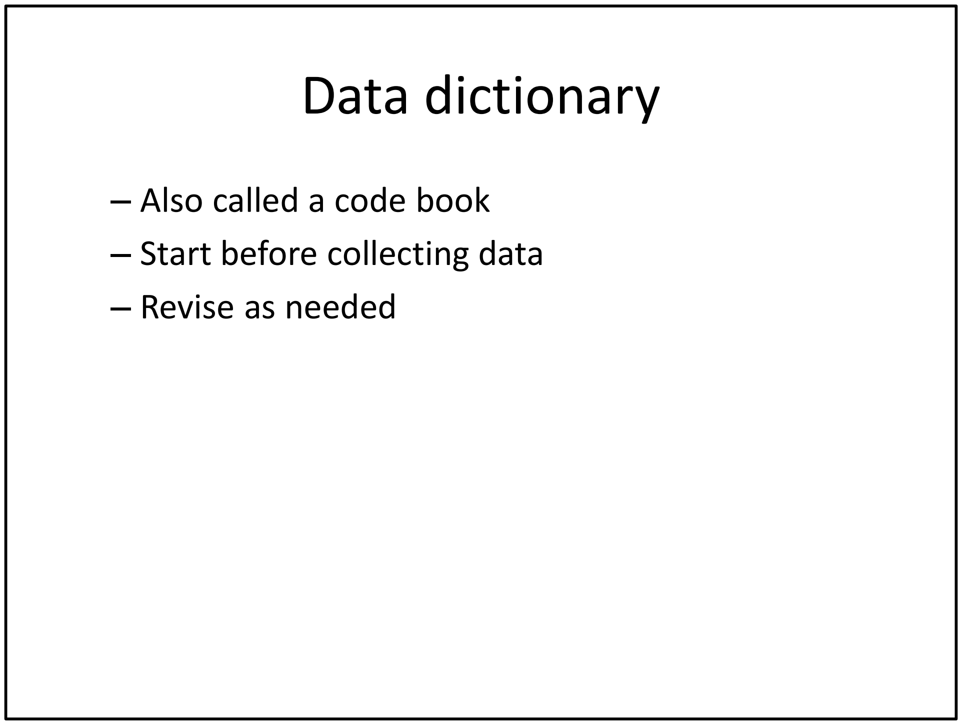


Data management

- Data dictionary
  - Variable names, Variable labels, Value labels, Missing value codes
- Managing complex files
  - Multiple response, Longitudinal/repeated measures data
- Storage options
  - Spreadsheet, Text file, Database, REDCap

There's a lot to cover in this video. I'll start with the data dictionary, which provides documentation on variable names, variable labels, value labels, and missing value codes. Then I want to mention two tricky issues with data management: multiple response variables and longitudinal/repeated measures data.

Let me just mention how happy I am to talk about data management issues. You don't know how much I have suffered over the years because the people I work with have failed to do some very basic data management. If nothing else, I want to make sure that you don't cause similar pain to your research teams in the future. Listen well and heed all the advice I give here today. It is built on top of a mountain of suffering.



Data dictionary

– Also called a code book
– Start before collecting data
– Revise as needed

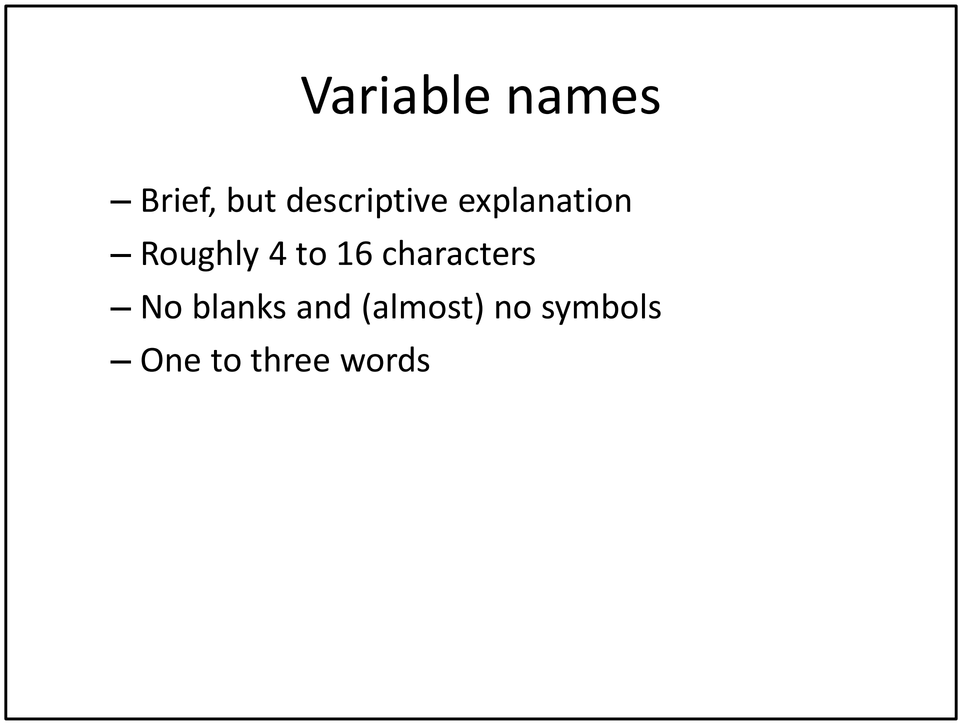
Your meta-data belongs in a data dictionary. The data dictionary, also known as the code book, is a short document (usually one page or less) that describes your data. It helps you organize and structure your data entry process. It also helps anyone else who might be working with you on the project. Finally, it is invaluable if you plan to share your data with outside groups.

If there is a distinction between a code book and a data dictionary, it is that the code book is usually more of an internal document where you tabulate rules about your data and decisions that you've had to make about unusual and unexpected situations.

An example of this is how to handle a five point Likert scale where someone circles both a 3 and a 4. Do you code that as a 3.5 or an invalid response? Either one is defensible, but you need to make sure that you write this down, so that the next time you encounter this situation, you'll make the choice that is consistent with your earlier action.

You should start a data dictionary before you collect any data. Update it during the data collection, if necessary.

I'll describe what belongs in a data dictionary after I describe the various elements of meta-data.

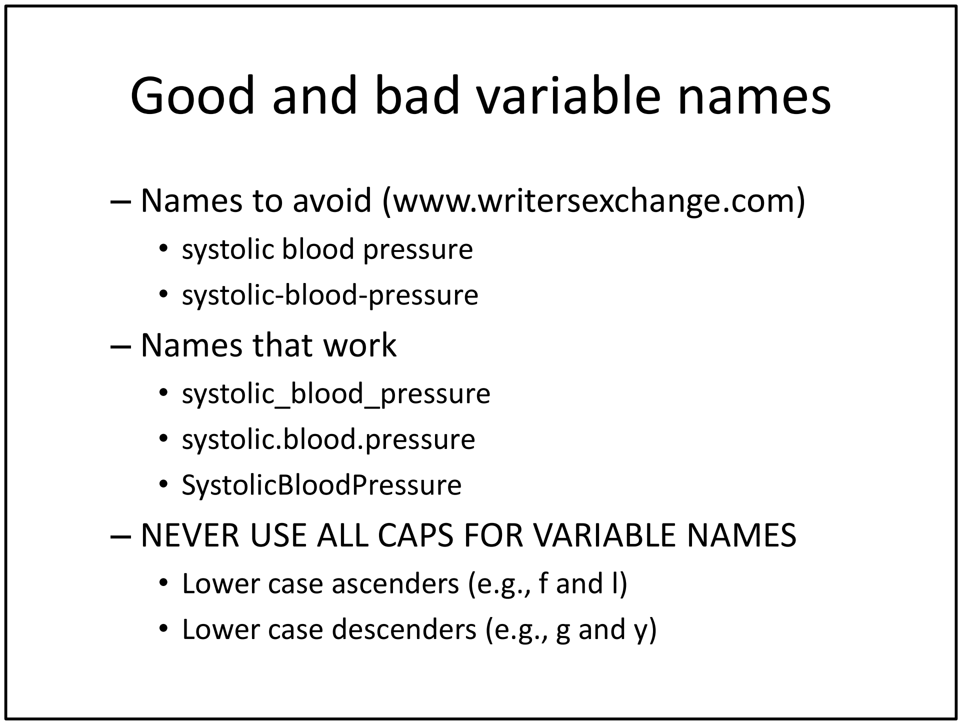


Variable names

- – Brief, but descriptive explanation
- – Roughly 4 to 16 characters
- – No blanks and (almost) no symbols
- – One to three words

Define a brief variable name for every variable that you collect. You will use these names when you analyze your data using a program like R, SAS, or SPSS. The variable name should be just long enough to be understandable, but short enough to minimize the typing in your data analysis program. I would recommend something between 4 to 16 characters and one or three short words.

Sometimes a very brief abbreviation is okay, like bw for birthweight or bmi for body mass index.



Good and bad variable names

- Names to avoid (www.writersexchange.com)
  - systolic blood pressure
  - systolic-blood-pressure
- Names that work
  - systolic_blood_pressure
  - systolic.blood.pressure
  - SystolicBloodPressure
- NEVER USE ALL CAPS FOR VARIABLE NAMES
  - Lower case ascenders (e.g., f and l)
  - Lower case descenders (e.g., g and y)

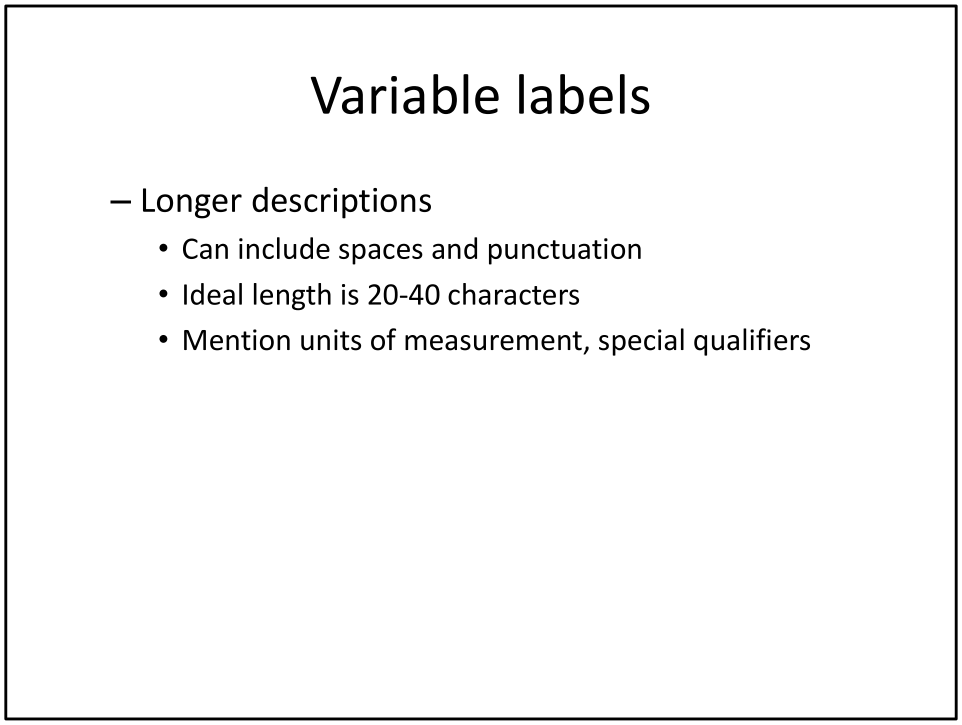
If your variable name is two or three short words (former smoker, or plasma beta carotene), you need to be careful. There's a story about a group called Writer's Exchange. They wanted to set up a website, www.writersexchange.com. It seemed like a good idea until they noticed that the website could be read as writer sex change.

You cannot use blanks between the words. It would work for some systems easily, for others, it would require some tedious work-arounds, and for still others, it would lead to system errors.

You can't connect the words with dashes, because most systems will interpret the dash as a subtraction. Actually, just about any symbol (dollar sign, percent sign, asterisk) is a no-no. This can seriously stifle your creativity, but you have to live with it. Most statistical software programs reserve these symbols for special purposes, and if you use them in your variable names, you will confuse things.

There are two exceptions, the underscore character and the dot. These can be used almost anywhere. The dot is tricky in SQL and Java, as I understand it, but fine almost everywhere else. I believe the underscore is fine just about anywhere. But you can also use a mixture of upper and lower case, known as CamelCase.

Please avoid the use of all upper case for variable names. It looks bad, but more importantly is harder to read. Lower case has the advantage of ascenders, letters that creep into the upper zone, like f and l, and descenders, letters that creep into the lower zone, like g and y. These make it easier to read lower case quickly and accurately. So either use all lower case like e.e. cummings, or mixed case with only the initial letter of a word capitalized.



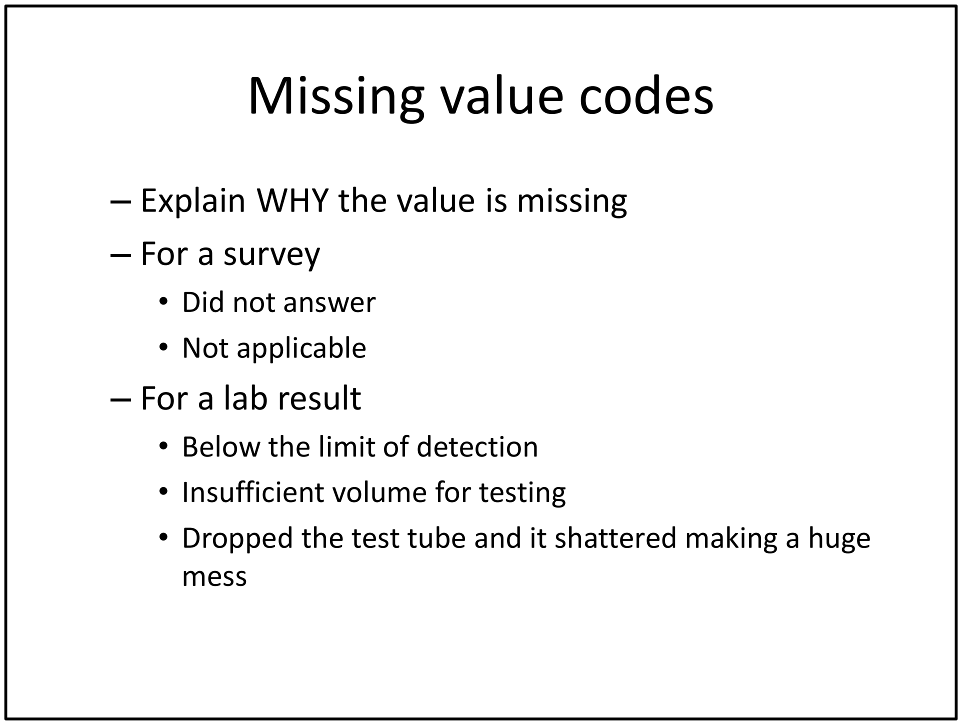
Variable labels

– Longer descriptions
  - Can include spaces and punctuation
  - Ideal length is 20-40 characters
  - Mention units of measurement, special qualifiers

Most statistical software programs have variable labels. R, sadly, is an exception. Variable labels are intended to be longer text fields that can describe your variables in greater detail. They typically have no limits on spaces, punctuation, or special characters. The variable labels will appear in your output in place of the variable name, making your output easier to follow.

You want to be more expansive with your variable labels: about 20 to 40 characters in length is good. Too much longer and the labels won't fit well on your tables or graphs.

If your variable has units of measurement, here is where you specify grams, kilograms, or pounds. If there are special qualifiers, like whether body temperature was measured orally, rectally, or under the axilla, here is where you put those details.



Missing value codes

– Explain WHY the value is missing
– For a survey
  - Did not answer
  - Not applicable
– For a lab result
  - Below the limit of detection
  - Insufficient volume for testing
  - Dropped the test tube and it shattered making a huge mess

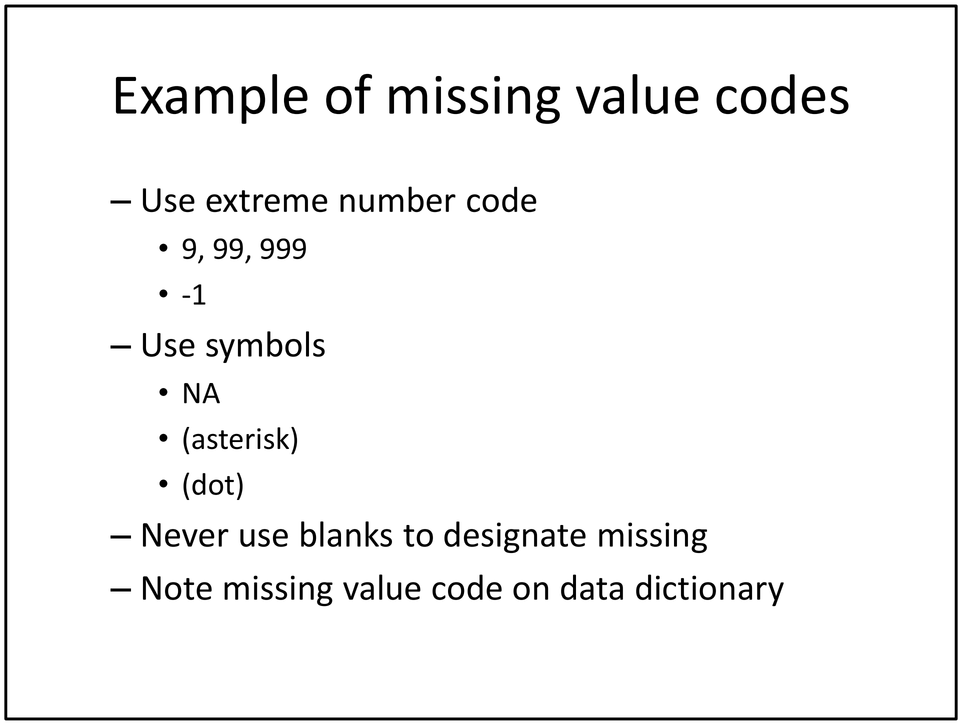
Sometimes you do not have data for a particular variable for some of your patients. This is perfectly fine. You must explain why the value is missing.

Sometimes a value is missing because a patient skipped the item on your questionnaire. Sometimes, though, you are the one who told the patient to skip the question. You were asking about the age of the patient's spouse and you thought that would be a difficult question for someone unmarried to answer. You should probably analyze those two different missing values quite differently.

If you are looking at lab results on a serum sample, there are several reasons why the lab result might be missing. Maybe the value was below the limit of detection. Maybe you did not have enough blood to test for this particular value. Maybe you dropped the test tube before you could analyze anything.

If you decided that you would replace the missing values with the average of the values where you did have data, that's not such a bad thing to do to fix things from the dropped test tube. There's no hint that a large or small value on the lab value would make the test tube more slippery. I'd worry a bit more about low sample volume. But the missing value where you absolutely cann ont and should not replace the missing value with the average of the remaining values is the case where the value was missing because it was below the limit of detection. Below low the limit of detection makes this particular lab value smaller than any of the others in the data set. And you’re going to replace it by an average? For this case, a simple solution is to replace the missing value with a value halfway between zero and the limit of detection.

That’s whay the reason for missingness is so important.



Example of missing value codes

- Use extreme number code
  - 9, 99, 999
  - -1
- Use symbols
  - NA
  - (asterisk)
  - (dot)
- Never use blanks to designate missing
- Note missing value code on data dictionary

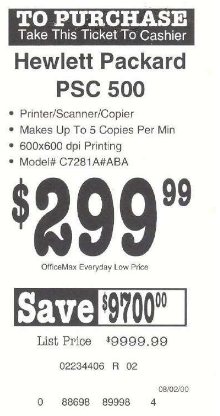
There are several ways you can code missing values. I recommend an extreme number code. 9 works well for values that can't get any larger than 4 or 5. Use 99 or 999 if the values can get larger. You want to choose enough nines so that the value is clearly outside of any reasonable range of legitimate values. So a birthweight of 99 kilograms is obviously a missing value for anyone except for a baby elephant. An IQ score of 999 is outside the range even of Albert Einstein.

Some people use -1 for a missing value code if the variable is clearly not ever going to be a negative value. A baby could not have a birth weight of -1 kilogram, because such a baby would float up to the ceiling after being delivered.

If there are several reasons why a variable could be missing, use a separate code for each reason, such as 997 for lost test tube, 998 for insufficient volume for testing, and 999 for a result below the limit of detection.

You can also use symbols for missing values such as an asterisk or a dot. You can also use the letters NA, which is what R expects. These last three have a disadvantage for numeric variables. They end up confusing some imports and force your numeric column into a character string.

Missing value example


TO PURCHASE
Take This Ticket To Cashier
Hewlett Packard
PSC 500
- Printer/Scanner/Copier
- Makes Up To 5 Copies Per Min
- 600x600 dpi Printing
- Model# C7281A#ABA

$299 99
OfficeMax Everyday Low Price
Save $9700 00
List Price $9999.99
02234406 R 02
08/02/00
0 88698 89998 4

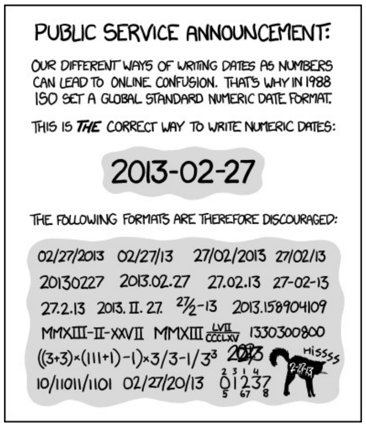
Price tage from computer store

Here's a humorous example of failure to account properly for missing values. The printer had a sales price of 299 dollars and 99 cents. The original price was listed as 9,999 dollars and 99 cents, which is an outrageous price for a printer. Clearly, a field of all nines was intended to be a missing value code. But the system that computed the savings did not recognize this as a missing value and calculated that if you bought this printer, you would be saving $9,700 dollars.

First break

- What have you learned
  - variable names
  - Variable labels
  - Missing value codes
- What's coming up
  - Date formats
  - Categorical values

Date formats


PUBLIC SERVICE ANNOUNCEMENT:
OUR DIFFERENT WAYS OF WRITING DATES AS NUMBERS CAN LEAD TO ONLINE CONFUSION. THAT'S WHY IN 1988 ISO SET A GLOBAL STANDARD NUMERIC DATE FORMAT.
THIS IS THE CORRECT WAY TO WRITE NUMERIC DATES:
2013-02-27
THE FOLLOWING FORMATS ARE THEREFORE DISCOURAGED:
02/27/2013 02/27/13 27/02/2013 27/02/13
20130227 2013.02.27 27.02.13 27-02-13
27.2.13 2013. II. 27. 27/2-13 2013.158904109
MMXIII-II-XXVII MMXIII LVII/CCCLXV 1330300800
((3+3)×(111+1)-1)×3/3-1/3³ 2013
HISSSSS
10/11011/1101 02/27/20/13 01237

Cartoon showing variety of data formats

As this cartoon shows, there are a variety of ways to code dates. This is Scott Munro, creator of the wonderful xkcd cartoon series. He shows the dates in a variety of formats that get more and more bizarre, including Roman numeral and binary versions of the date. He is trying to emphasize the importance of following a single standard. He recommends, and I heartily agree with his recommendation, that you should always use the ISO 8601 standard.

The ISO 8601 standard for dates starts with the four digit year followed by a dash. Then you have two digits for the month. It is always two digits, so January is 01 rather than 1 and September is 09 rather than 9. This is followed by another dash and then the two digit day. Note that the first day of the month is 01 rather than 1.

One nice side effect of the ISO 8601 standard is that if the date gets converted to a string, that string can be sorted and it will always sort by the year, then the month, and then the date, which is exactly what you want.

Nobody follows this standard. The most common deviation is failing to put a zero in front of single digit months and single digit dates. Leaving out that zero ruins the ability to sort the data as I noted early because the months will be sorted as 1, 10, 11, 12, then 2, 3, 4, etc.

Another deviation is using slashes rather than dashes. This is a minor issue, but a much bigger problem is changing the year, month, date order. In America, we like to put the month, followed by the day, followed by the year. In Europe, they put the day followed by the month, followed by the year. Note only does this ruin the sort order, but dates like 10/12/2019 are ambiguous. Is it October 12 or December 10?

Sometimes you will see the month spelled out or with a three letter abbreviation. This also ruins the sort order, as April is not the first month of the year, even though it appears first alphabetically. It also causes problems internationally, since the names of the months are differnt in different languages.

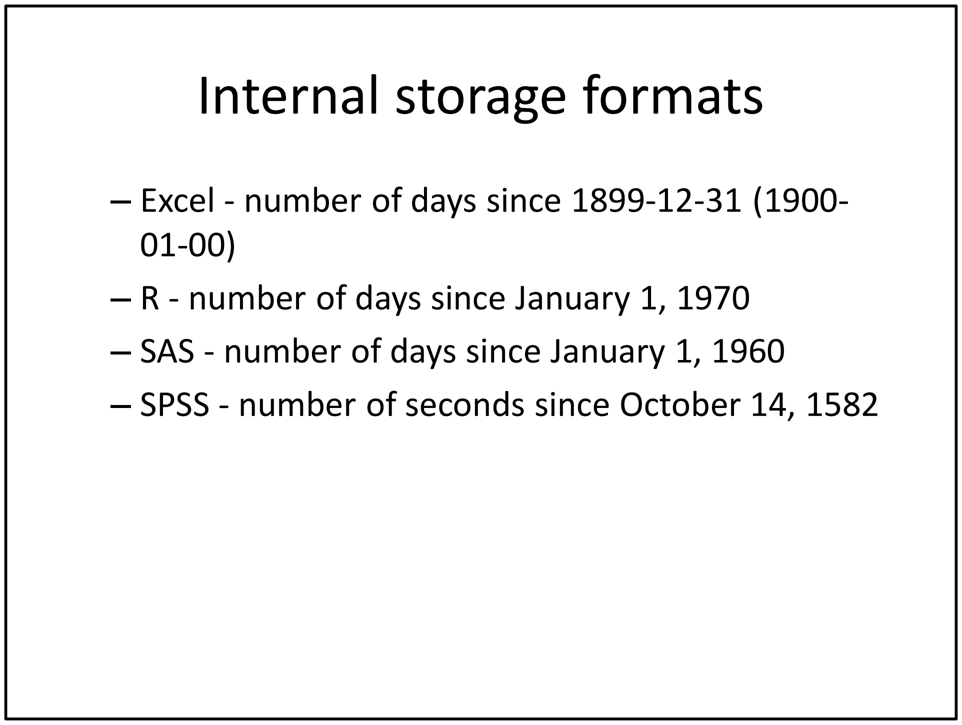
Does it seem like I'm nitpicking. Well maybe I am. But here is the one thing that I hope you will agree is a major issue and not a nitpick.

Sometimes, people only use two digits for the year. This is a serious problem and it leads to little kids being classified as 101 years old.

Do any of you remember the Y2K crisis? It was a big bust, but let me tell you about my two plans to profit from the Y2K crisis. My first plan was to go to the ATM machine at 11:59pm on December 31 and withdraw the maximum amount in the hopes that the ATM would crash one minute later and lose the record of my transaction. My second plan was to show up late for work on the Monday following Y2K and explain to my boss that I was only late because my alarm clock was not Y2K compliant.

It turns out that the concerns about Y2K did not materialize. Either because they were overblown or because all the programming that we did to fix things actually saved ourselves from a major computer meltdown. But the Y2K crisis still affects data analysis today.

Never, never, never code your data as two digit years. Even if all the dates are year 2000 and later. You are asking for trouble if you don't use four digit years.



Internal storage formats

- Excel - number of days since 1899-12-31 (1900-01-00)
- R - number of days since January 1, 1970
- SAS - number of days since January 1, 1960
- SPSS - number of seconds since October 14, 1582

Everything in the computer is stored as a number. That includes dates. Your computer is smart enough to turn that number into a string that you can interpret properly, but the underlying storage has to be a number so that your computer can calculate things like the number of days between when you entered the hospital and when you left.

Every system has a different way of storing dates and you can summarize this best by noting what the index date is, the date that is stored internally as a zero.

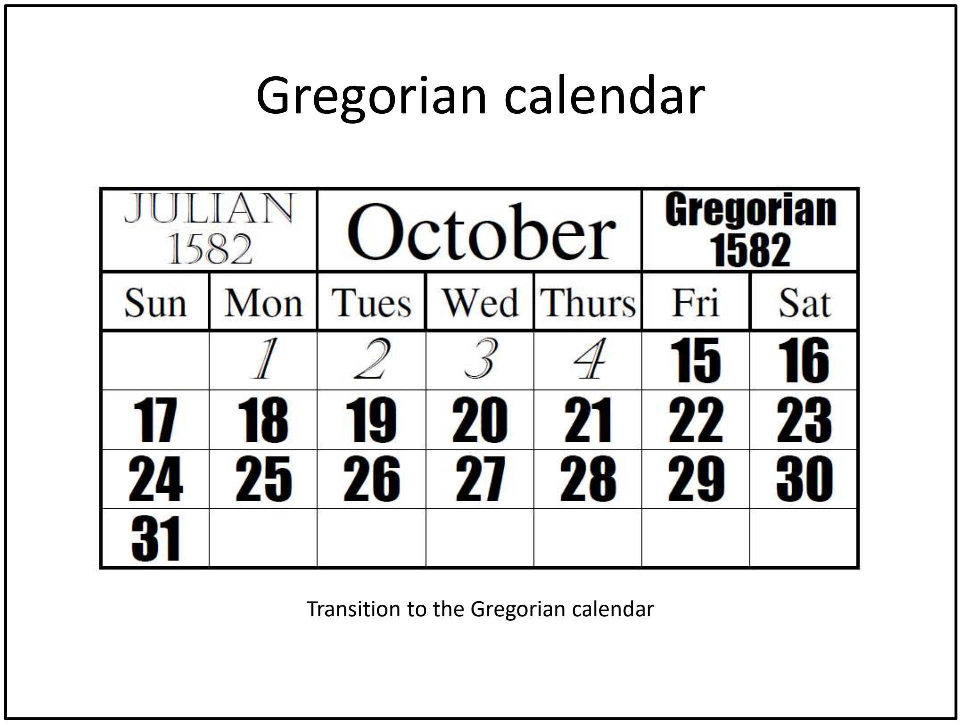
Every program has a different index data and this can sometimes lead to problems when you switch from one program to another.

Excel decided to make their index date the start of the 20th century, which seems simple enough, but they made two mistakes. First, they assigned zero to a non-existent date (January 0, 1900). You can fix this by considering the index date to be December 31, 1899. A bigger problem is the geniuses at Microsoft failed to recognize that 1900 was NOT a leap year.

For what it’s worth, Microsoft did it this way to make things compatible with another spreadsheet, Lotus 1-2-3. Also, some of the Microsoft Excel versions for the Macintosh used an index date of 1904-01-01. This was also done to maintain compatibility, this time with other Macintosh programs, since earlier versions of Macintosh refused to compute dates earlier than 1904.

That sounds like a nitpick, but having two different index dates and making leap year mistakes can lead to conversion problems. The moral is to always do some spot checking on dates when converting from Excel to a different system. Being off by four years from what you think the date is could really screw things up. Even being off by a single day because of a leap year screw up could mess up an inclusion/exclusion criteria based on the year of birth.

R and SAS choose more current index dates. The choice of an index date for SPSS has a more interesting story behind it.



Gregorian calendar

JULIAN 1582		October			Gregorian 1582	
Sun	Mon	Tues	Wed	Thurs	Fri	Sat
	1	2	3	4	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

Transition to the Gregorian calendar

if the date of October 14, 1582 seems strange to you, it represents the start of the Gregorian calendar. The Julian calendar developed in ancient times was quite impressive. Astonomers at the time knew that the earth took 365.25 days to travel one orbit around the sun. So they invented the concept of a leap year.

Unfortunately, the earth actually takes 365.24219 days. That 0.00781 seems small but it means that you are off but a full day every 128 years. The Church was becoming concerned because Easter, a holiday that clearly belong in the Spring was falling earlier and earlier in the season. The astronomers figured out that 3/400 or 0.0075 was pretty close, so if they could find a way to shorten the calendar by 3 days every 400 years, they'd keep things well in sync. So they decided to toss out leap years on three of the four century years. That would make 1700, 1800, and 1900 normal years, but 2000 would still be a leap year.

Now there is even a discrepancy between (3/400) 0.0075 and 0.00781 so the proposed solution is to take back another leap day in the year 4000 and then again in the year 8000. I hope to be alive to see these two events.

And to make up for all the accumulated errors over the past fifteen centuries, they decided to remove ten days from the calendar. October 4, 1582 would be followed by October 15, 1582, putting Easter back to mid-Spring, where it properly belonged for then and for all future generations. So the skipped over date of October 14, 1582 is day 0 of the Gregorian calendar.

Here is an illustration of the transition, thanks to By Asmdemon - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=35391210

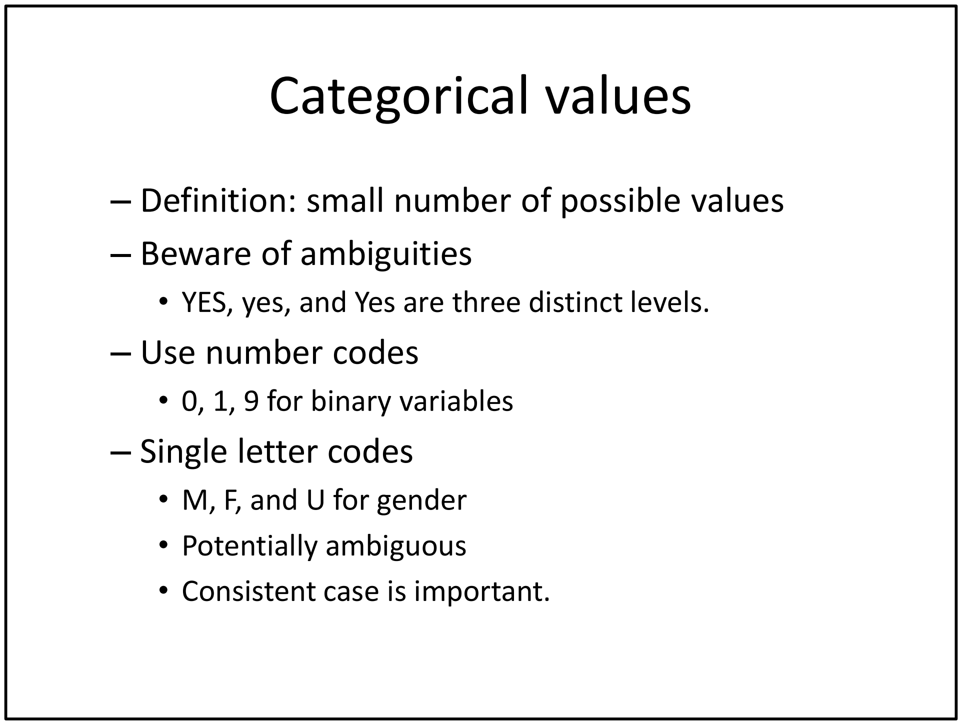
Gregorian calendar

Painting of Pope Gergory XIII

Here’s a picture of Pope Gregory XIII, taken from Wikipedia. He’s the Pope responsible for the Gergorian calendar. For what it’s worth, it took a lot of time for some countries to convert to the Gregorian calendar. England (and the American colonies) switched in 1752 and because of their delay had to miss eleven days and not ten. They would have switched earlier, but did not like being dictated to by the Pope. Countries where the Orthodox Church predominated were even slower. Russia did not switch until the Communists took over in 1918 and Greece waited until 1923.

Enough of the trivia. Let’s get back to real hard data.

So why did SPSS choose the number of seconds since October 14, 1582 rather than the number of days. Perhaps they were concerned about some of the rounding that could occur when you were tracking dates and times and not just dates. if you count by the number of elapsed seconds, you never have to round. Now, in exchange, you have to deal with date values that are on the order of 13 billion, and if you do a length of stay calculation, you are calculating the number of seconds that you were stuck in the hospital and not the number of days you were stuck in the hospital. You have to divide by 86,400 (the number of seconds in a day) to get length of stay in days.



Categorical values

- Definition: small number of possible values
- Beware of ambiguities
  - YES, yes, and Yes are three distinct levels.
- Use number codes
  - 0, 1, 9 for binary variables
- Single letter codes
  - M, F, and U for gender
  - Potentially ambiguous
  - Consistent case is important.

A categorical variable is a variable with a small number of possible values. What’s small? If you can state all the possible values without taking a breath, that’s categorical. Actually, don’t worry excessively about what meets the definition of categorical. Use a bit of common sense and you’ll be fine.

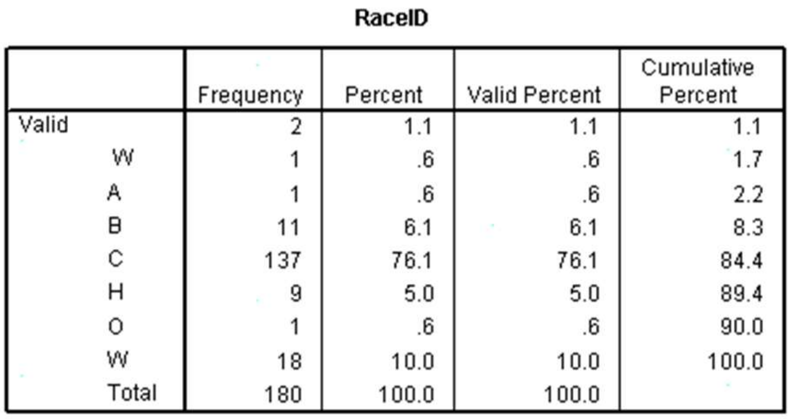
Some people use text strings to designate categorical variables. This requires more typing, which may not be too serious a concern, but it also raises the possibility of ambiguity, especially if several people are responsible for data entry.

Note that the strings YES (all upper case), yes (all lower case), and Yes (Mixed case) may look the same to you, but they are treated as separate categories by the computer.

Your book recommends number codes, and I agree with them. But you should know that others hate number codes and prefer the use of short strings or single letter codes.

Single letter codes are easy to remember. The letter codes (M, F, and U) for gender are obviously Male, Female, and Unknown. These single letters are also convenient plotting codes. At times, the codes can be ambiguous. Did you code your patients W for white, or C for Caucasian?

Example of ambiguous coding



RaceID

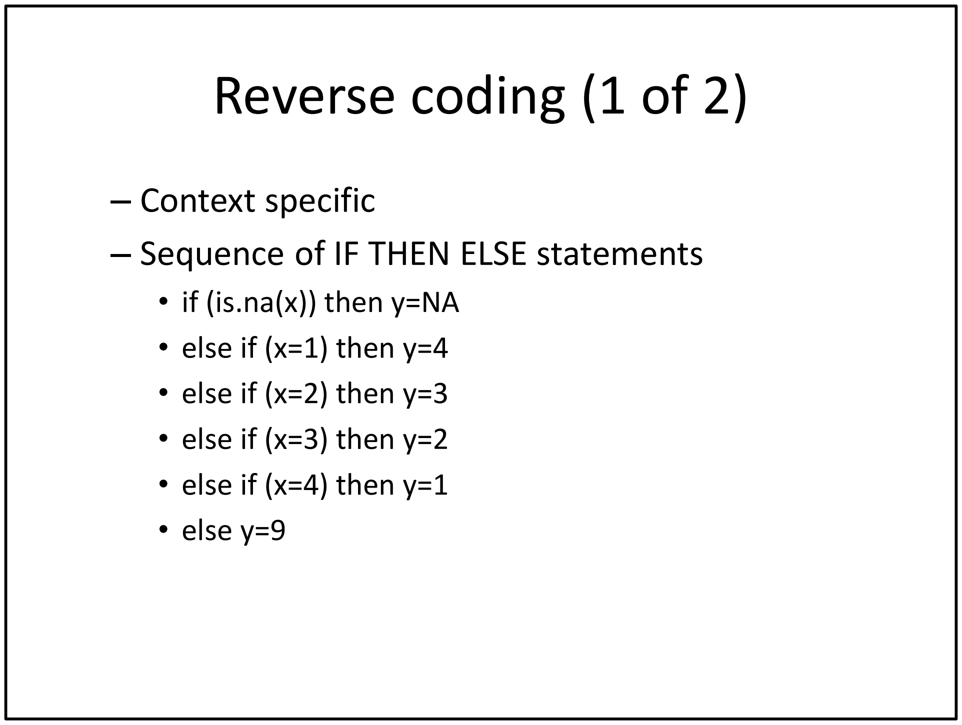
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid		2	1.1	1.1	1.1
	W	1	.6	.6	1.7
	A	1	.6	.6	2.2
	B	11	6.1	6.1	8.3
	C	137	76.1	76.1	84.4
	H	9	5.0	5.0	89.4
	O	1	.6	.6	90.0
	W	18	10.0	10.0	100.0
	Total	180	100.0	100.0	

SPSS frequencies table for RaceID

Here is an example of ambiguous data that I documented on my website back in 2007. I had a data set with a variable RaceID, and when I examined the data using the frequencies procedure in SPSS, I got this table. It shows two observations with blank values at the very top. This is bad form, as I noted earlier, and a special code letter like U or X should be presented instead. Notice also that W appears twice. The first W has a slight indent to it, which gives you a hint that something is not quite right. It turns out that there was a patient with a race code that was two characters long rather than one character. And those two characters were blank and W.

That was easy enough to fix, but this is why they pay me the big bucks. Because I pay very close attention to small details that might otherwise screw up the analysis.

One more nitpick. The code for A is ambiguous. Does it mean Asian American or American Indian? I can guess most of the others: B for Black, C for Causasian, H for Hispanic, O for Other and W for White (Why was this not combined with C?).

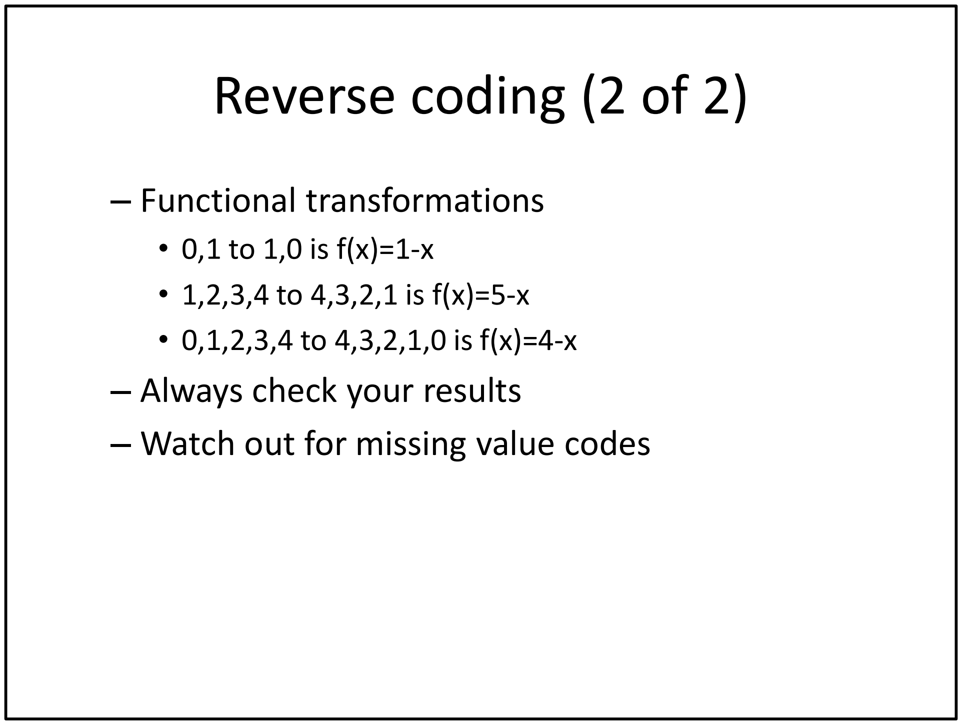


Reverse coding (1 of 2)

- Context specific
- Sequence of IF THEN ELSE statements
  - if (is.na(x)) then y=NA
  - else if (x=1) then y=4
  - else if (x=2) then y=3
  - else if (x=3) then y=2
  - else if (x=4) then y=1
  - else y=9

Reverse scaling is tedious and always go slowly, because tedium tends to produce simple mistakes. SPSS has a very nice recode dialog box that is hard to mess up.

You can also use a series of IF THEN ELSE commands to do the reverse coding. Always start by recoding the missing value. Notice that I took the time to code as 9 anything outside what I believed were the only possible values.



Reverse coding (2 of 2)

- Functional transformations
  - 0,1 to 1,0 is f(x)=1-x
  - 1,2,3,4 to 4,3,2,1 is f(x)=5-x
  - 0,1,2,3,4 to 4,3,2,1,0 is f(x)=4-x
- Always check your results
- Watch out for missing value codes

But if you want to be clever, there is usually a simple mathematical function of the form k-x or k+1-x that will reverse scale everything all at once.

If you had a variable with values 0 and 1, the function f(x)=1-x will effectively convert the 0 to a 1 and the 1 to a 0.

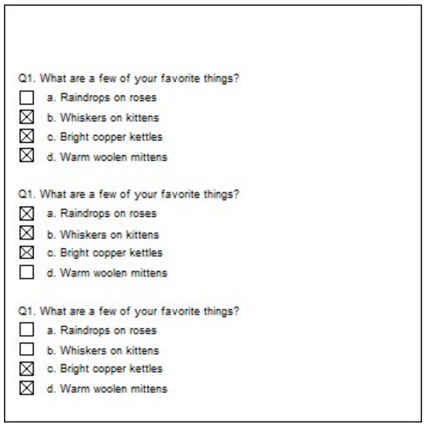
If you had a variable with values 1, 2, 3, and 4, the function f(x)=5-x would effectively reverse these values. But note that if 0 were legal, if the possible values were 0, 1, 2, 3, and 4, then the proper function would be f(x)=4-x.

Always account properly for missing values when reverse scaling. It's a good idea to do a simple frequency table before and after reverse coding. It is also a good idea to store the reverse coded variable under a different name, so that you could easily undo any mistakes that you might have made.

Second break

- What have you learned
  - Dates
  - Value labels
  - Reverse coding
- What's coming next
  - Multiple response
  - Longitudinal/repeated measures data

A multiple response example



Q1. What are a few of your favorite things?
☐ a. Raindrops on roses
☒ b. Whiskers on kittens
☒ c. Bright copper kettles
☒ d. Warm woolen mittens

Q1. What are a few of your favorite things?
☒ a. Raindrops on roses
☒ b. Whiskers on kittens
☒ c. Bright copper kettles
☐ d. Warm woolen mittens

Q1. What are a few of your favorite things?
☐ a. Raindrops on roses
☐ b. Whiskers on kittens
☒ c. Bright copper kettles
☒ d. Warm woolen mittens

Questionnaire with a multiple response question

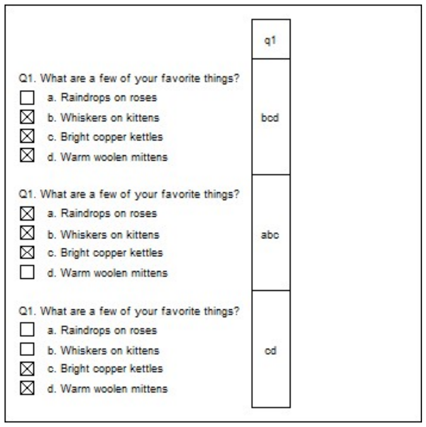
This is an example of a questionnaire where the patients can choose one of more of the following favorite things: Raindrops on roses, or whiskers on kittens. Bright copper kettles, or warm woolen mittens.

The first patient liked the last three things: whiskers on kittens, bright copper kettles, and warm woolen mittens.

The second patient liked the first three things: raindrops on roses, whiskers on kittens, and bright copper kettles.

The third patient liked the last two things: bright copper kettles, and warm woolen mittens.

Coding multiple response with a single column

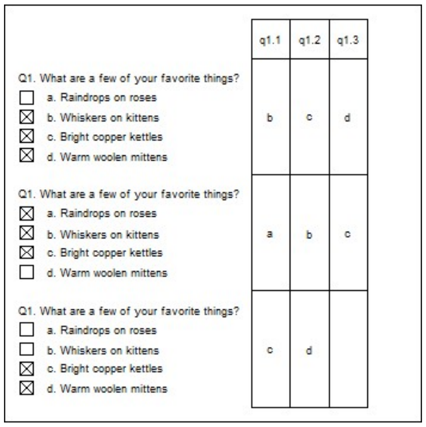


Multiple response coded into a single column

You can code all the possible responses in a single column. Since the first person chose the last three responses, code their response as bcd. The second person chose the first three responses, and would be coded as abc. The third person only chose the last two responses, and would be coded cd.

This approach makes it difficult to compute even simple statistics, such as the proportion of patients who chose "Whiskers on kittens." You'd have to work with a string search function. How many of the strings contain the letter b? It's not a terribly hard amount of work, but with a bit of planning, you could make it a bit easier for yourself down the road.

A different way to code multiple response



Q1. What are a few of your favorite things?
- ☐ a. Raindrops on roses
- ☒ b. Whiskers on kittens
- ☒ c. Bright copper kettles
- ☒ d. Warm woolen mittens

Q1. What are a few of your favorite things?
- ☒ a. Raindrops on roses
- ☒ b. Whiskers on kittens
- ☒ c. Bright copper kettles
- ☐ d. Warm woolen mittens

Q1. What are a few of your favorite things?
- ☐ a. Raindrops on roses
- ☐ b. Whiskers on kittens
- ☒ c. Bright copper kettles
- ☒ d. Warm woolen mittens

q1.1	q1.2	q1.3
b	c	d
a	b	c
c	d	

A multiple response question coded into three columns

You could split this into three columns, putting the first selection in the first column, the second selection in the second column, and the third selection in the third column. So the bcd gets split into b in the first column, c in the second column, and d in the third column. The abc gets split into a in the first column, b in the second column, and c in the third column. The cd gets split into c in the first column, d in the second column and the third column is left blank.

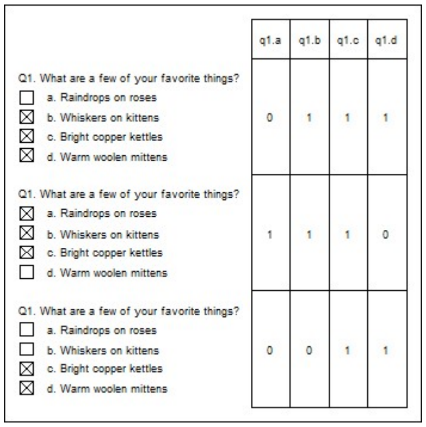
This is a bit better, but there are still some problems.

This format produces some unneeded missing values, such as for the patient who has less than three favorite things.

If you find out halfway through the survey that someone chose all four favorite things, you'd have to go back and add a new column.

This format is also a bit tricky. Notice how even though all three patients selected c, the letter appears once in the third column, once in the second column, and once in the first column. If you wanted the percentage of patients who chose "c," you'd have to calculate the percentage of c's in the first column, and add it to the percentage of c's in the second column and the percentage of c's in the third column.

The recommended way to code multiple response



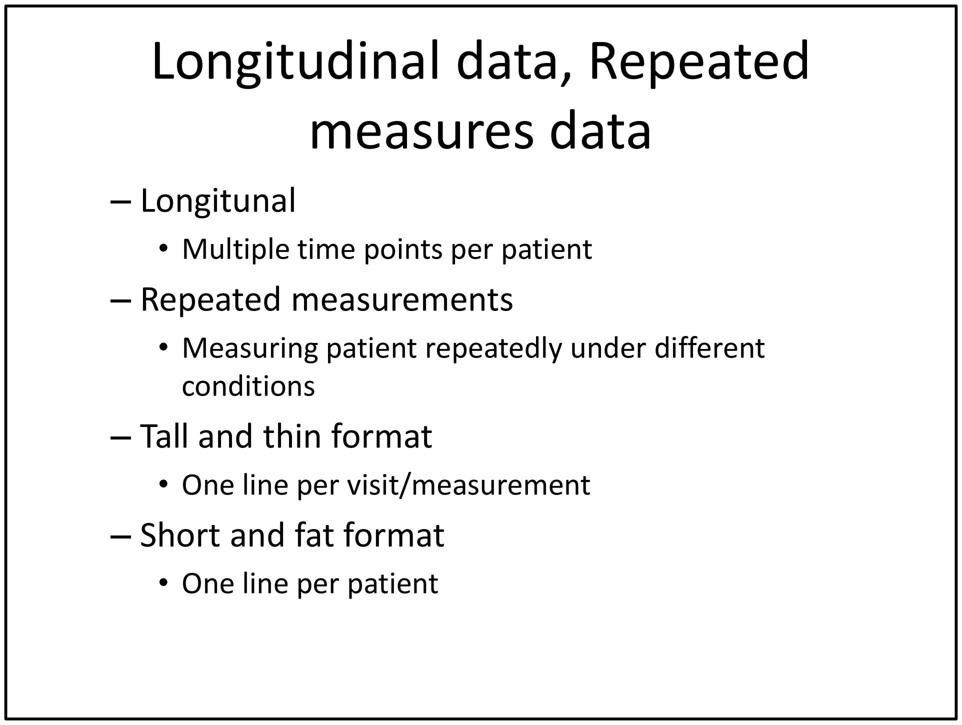
	q1.a	q1.b	q1.c	q1.d
Q1. What are a few of your favorite things? ☐ a. Raindrops on roses ☒ b. Whiskers on kittens ☒ c. Bright copper kettles ☒ d. Warm woolen mittens	0	1	1	1
Q1. What are a few of your favorite things? ☒ a. Raindrops on roses ☒ b. Whiskers on kittens ☒ c. Bright copper kettles ☐ d. Warm woolen mittens	1	1	1	0
Q1. What are a few of your favorite things? ☐ a. Raindrops on roses ☐ b. Whiskers on kittens ☒ c. Bright copper kettles ☒ d. Warm woolen mittens	0	0	1	1

Multiple response coded with individual item indicators

The recommended way to code multiple response is to create indicator variables for each possible response. The q1.a column is one for the second patient, as he/she was the only one who who liked raindrops on roses. The other two patients get a zero in this column. The q1.b column is one for the first and second patients, who both liked whiskers on kittens, but zero for the third patient who did not like whiskers on kittens. What's wrong here? Who doesn't like whiskers on kittens? The q1.c column is one for everybody because everybody liked bright copper kettles. Finally, in column q1.d, the first and third patients get ones for the like of warm woolen mittens, and the second patient gets a zero.

So the number of patients who like whiskers on kittens is just the number of ones in column q1.b.

Also note another big advantage here–no blank cells!



Longitudinal data, Repeated measures data

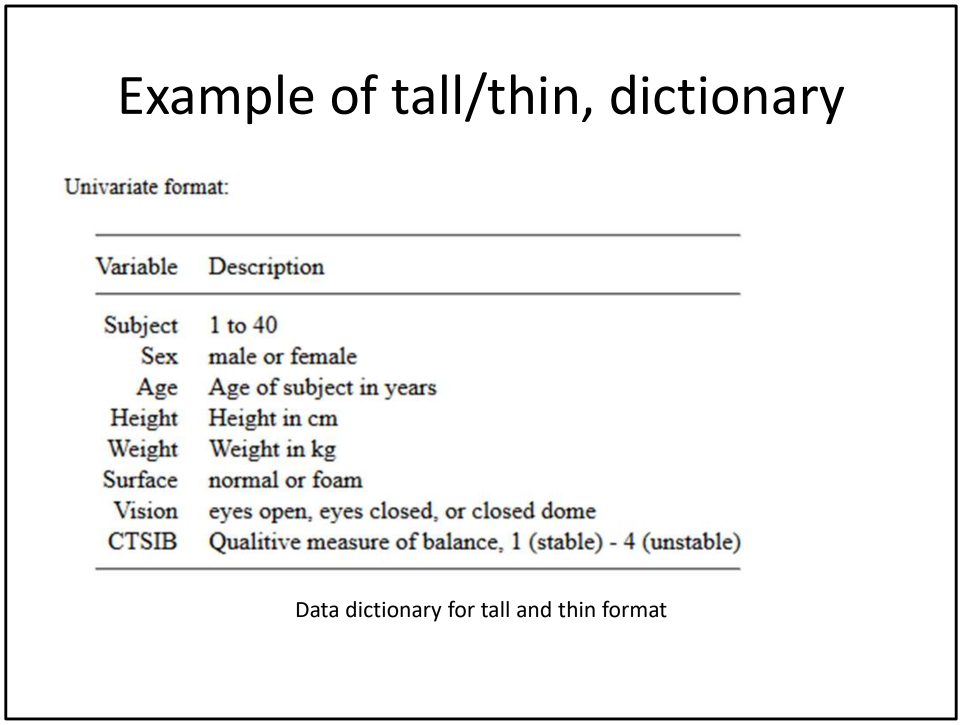
- Longitunal
  - Multiple time points per patient
- Repeated measurements
  - Measuring patient repeatedly under different conditions
- Tall and thin format
  - One line per visit/measurement
- Short and fat format
  - One line per patient

Longitudinal data are data collected at several different time periods on the same individual. Repeated measurements data are the same measurements collected on an individual under different conditions. Both longitudinal and repeated measurements data are very powerful because each subject can serve as their own control. Data analysis is tricky, but data management is even trickier.

You have two different options for storing longitudinal or repeated measurements data. The tall and thin format has one row for each visit in a longitudinal study or one row for each measurement in a repeated measures study. This means lots of rows, but not so many columns. This makes the data set tall and thin.

The short and fat format has one row per patient and each visit or each repeated measure is spread out across multiple variables. This means fewer rows but many more columns.

Some data analyses, such as plots, work better with the tall and thin format. Other analyses, such as correlations, work better with the short and fat format. It is tedious to switch between the two formats, but you can do this.



Example of tall/thin, dictionary

Univariate format:

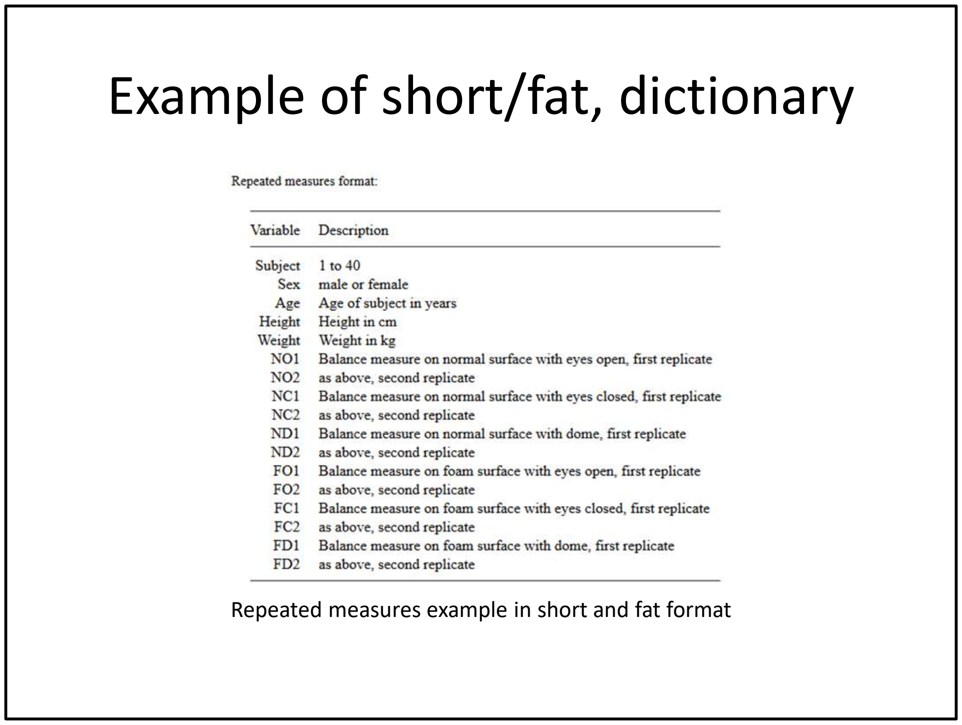
Variable	Description
Subject	1 to 40
Sex	male or female
Age	Age of subject in years
Height	Height in cm
Weight	Weight in kg
Surface	normal or foam
Vision	eyes open, eyes closed, or closed dome
CTSIB	Qualitive measure of balance, 1 (stable) - 4 (unstable)

Data dictionary for tall and thin format

Here's a data set looking at balance. Patients are asked to balance on one foot either on a normal surface or on a softer surface of foam. I presume that balancing is harder on a soft surface like foam. They do this under three vision conditions: eyes open, eyes closed, and a closed dome which allows a little bit of partial vision. I presume that balancing is easier with eyes open, hardest with eyes colses, and somewhere in the middle for a closed dome. Each combination of surface and vision was repeated twice, leading to 2 by 3 by 2 or 12 different repeated measurements.

The outcome variable is an observer assessment of balance on a four point scale.

This is the data dictionary for the tall and thin formatted data. A single column holds the observer assessment of balance, which means 12 by 40 or 480 rows of data.



Example of short/fat, dictionary

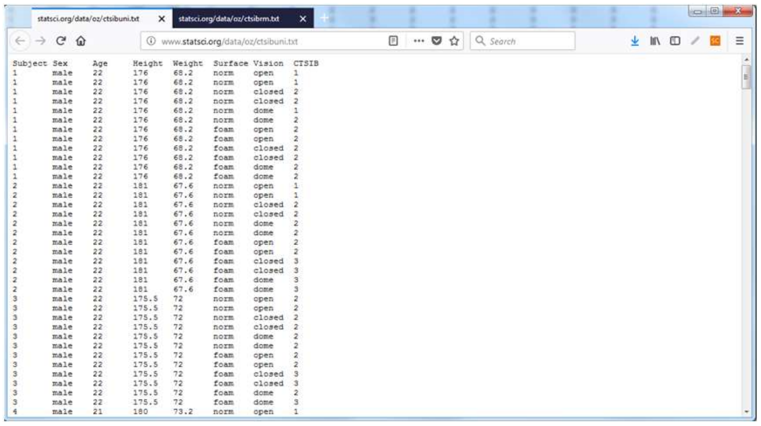
Repeated measures format:

Variable	Description
Subject	1 to 40
Sex	male or female
Age	Age of subject in years
Height	Height in cm
Weight	Weight in kg
NO1	Balance measure on normal surface with eyes open, first replicate
NO2	as above, second replicate
NC1	Balance measure on normal surface with eyes closed, first replicate
NC2	as above, second replicate
ND1	Balance measure on normal surface with dome, first replicate
ND2	as above, second replicate
FO1	Balance measure on foam surface with eyes open, first replicate
FO2	as above, second replicate
FC1	Balance measure on foam surface with eyes closed, first replicate
FC2	as above, second replicate
FD1	Balance measure on foam surface with dome, first replicate
FD2	as above, second replicate

Repeated measures example in short and fat format

Here's the data dictionary for the short and fat format. There's one row per patient, but this means that the twelve repeated measures conditions have to have separate columns. NO1 and NO2 represent the normal surface, open eyes measurements. NC1 and NC2 represent the normal surface, closed eyes measurements. The last two columns, FD1 and FD2 represent a foam surface and domed vision. I'll leave it to you to decode the middle values.

Example of tall/thin, data



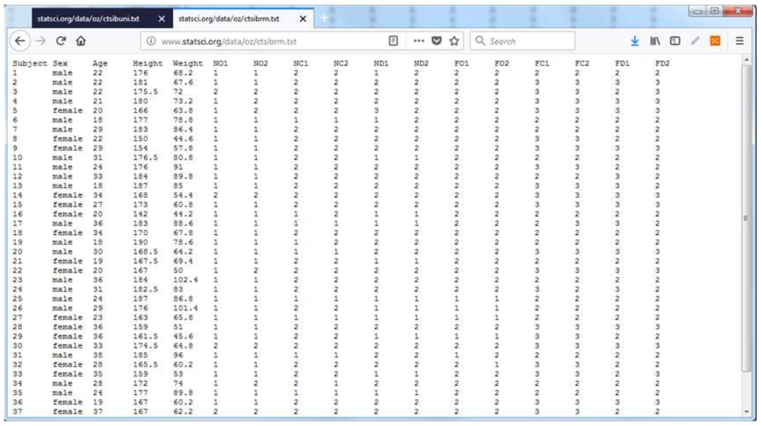
```
Subject Sex     Age     Height  Weight  Surface Vision  CTSIB
1       male    22      176     68.2    norm    open    1
1       male    22      176     68.2    norm    open    1
1       male    22      176     68.2    norm    closed  2
1       male    22      176     68.2    norm    closed  2
1       male    22      176     68.2    norm    dome    1
1       male    22      176     68.2    norm    dome    2
1       male    22      176     68.2    foam    open    2
1       male    22      176     68.2    foam    open    2
1       male    22      176     68.2    foam    closed  2
1       male    22      176     68.2    foam    closed  2
1       male    22      176     68.2    foam    dome    2
1       male    22      176     68.2    foam    dome    2
2       male    22      181     67.6    norm    open    1
2       male    22      181     67.6    norm    open    1
2       male    22      181     67.6    norm    closed  2
2       male    22      181     67.6    norm    closed  2
2       male    22      181     67.6    norm    dome    2
2       male    22      181     67.6    norm    dome    2
2       male    22      181     67.6    foam    open    2
2       male    22      181     67.6    foam    open    2
2       male    22      181     67.6    foam    closed  3
2       male    22      181     67.6    foam    closed  3
2       male    22      181     67.6    foam    dome    3
2       male    22      181     67.6    foam    dome    3
3       male    22      175.5   72      norm    open    2
3       male    22      175.5   72      norm    open    2
3       male    22      175.5   72      norm    closed  2
3       male    22      175.5   72      norm    closed  2
3       male    22      175.5   72      norm    dome    2
3       male    22      175.5   72      norm    dome    2
3       male    22      175.5   72      foam    open    2
3       male    22      175.5   72      foam    open    2
3       male    22      175.5   72      foam    closed  3
3       male    22      175.5   72      foam    closed  3
3       male    22      175.5   72      foam    dome    2
3       male    22      175.5   72      foam    dome    3
4       male    21      180     73.2    norm    open    1
```

statsci.org/data/oz/ctsibuni.txt
statsci.org/data/oz/ctsibrm.txt
www.statsci.org/data/oz/ctsibuni.txt
Search

Repeated measures data in tall and thin format

Here’s the data for the tall and thin format. Don’t worry about individual values, but do notice the size of the scroll bar on the right hand side is quite small indicating that you are only seeing a small fraction of the available rows.

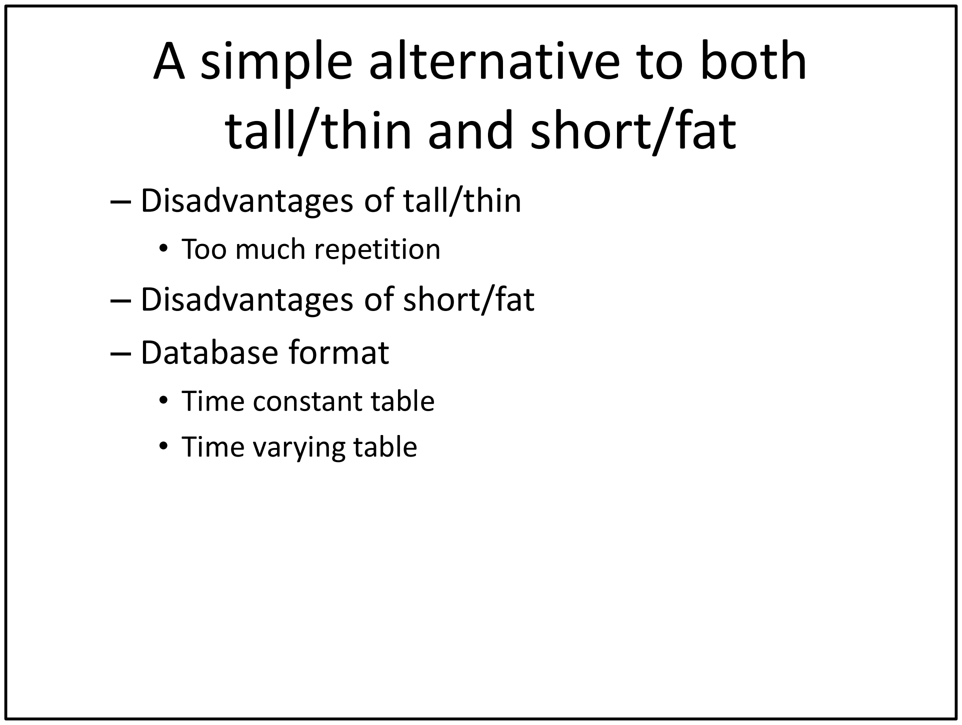
Example of short/fat, data


statsci.org/data/oz/ctsibuni.txt
statsci.org/data/oz/ctsibrm.txt
www.statsci.org/data/oz/ctsibrm.txt
Search

Subject	Sex	Age	Height	Weight	NO1	NO2	NC1	NC2	ND1	ND2	FO1	FO2	FC1	FC2	FD1	FD2
1	male	22	176	68.2	1	1	2	2	1	2	2	2	2	2	2	2
2	male	22	181	67.6	1	1	2	2	2	2	2	2	3	3	3	3
3	male	22	175.5	72	2	2	2	2	2	2	2	2	3	3	2	3
4	male	21	180	73.2	1	2	2	2	2	2	2	2	3	3	3	3
5	female	20	166	63.8	1	2	2	2	3	2	2	2	3	3	3	3
6	male	18	177	78.8	1	1	1	1	1	2	2	2	2	2	2	2
7	male	29	183	86.4	1	1	2	2	2	2	2	2	2	2	2	2
8	female	22	150	44.6	1	1	2	2	2	2	2	2	3	3	2	2
9	female	29	154	57.8	1	1	2	2	2	2	2	2	3	3	3	3
10	male	31	176.5	80.8	1	1	2	2	1	1	2	2	2	2	2	2
11	male	24	176	91	1	1	2	2	2	2	2	2	3	3	2	2
12	male	33	184	89.8	1	1	2	2	2	2	2	2	2	2	3	2
13	male	18	187	85	1	1	2	2	2	2	2	2	3	3	3	2
14	female	34	168	54.4	2	2	2	2	2	2	2	2	3	3	3	3
15	female	27	173	60.8	1	1	2	2	2	2	2	2	3	3	3	3
16	female	20	142	44.2	1	1	1	2	1	1	2	2	2	2	2	2
17	male	36	183	88.6	1	1	1	1	1	1	2	2	2	3	3	2
18	female	34	170	67.8	1	1	2	2	2	2	2	2	2	2	2	2
19	male	18	190	78.6	1	1	1	2	2	2	2	2	2	2	2	2
20	male	30	168.5	64.2	1	1	1	1	2	2	2	2	3	3	3	3
21	female	19	167.5	69.4	1	1	2	2	1	1	2	2	2	2	2	2
22	female	20	167	50	1	2	2	2	2	2	2	2	3	3	3	3
23	male	36	184	102.4	1	1	2	2	2	2	2	2	2	2	2	2
24	male	31	182.5	83	1	1	2	2	2	2	2	2	3	2	3	2
25	male	24	187	86.8	1	1	1	1	1	1	1	1	2	2	2	2
26	male	29	176	101.4	1	1	2	2	1	1	1	1	2	2	2	2
27	female	23	163	65.8	1	1	1	1	1	1	1	1	2	2	2	2
28	female	36	159	51	1	1	2	2	2	2	2	2	3	3	3	3
29	female	36	161.5	45.6	1	1	2	2	1	1	1	1	3	3	2	2
30	female	33	174.5	64.8	2	2	2	2	2	2	2	2	3	3	3	3
31	male	38	185	96	1	1	1	1	2	2	1	2	2	2	2	2
32	female	28	165.5	60.2	1	1	2	2	2	2	2	1	3	3	2	2
33	female	35	159	53	1	1	2	2	1	1	2	2	3	3	2	3
34	male	28	172	74	1	2	2	1	2	2	2	2	2	2	2	2
35	male	24	177	89.8	1	1	1	1	1	1	2	2	2	2	2	2
36	female	19	167	60.2	1	1	2	2	2	2	2	2	3	3	3	3
37	female	37	167	62.2	2	2	2	2	2	2	2	2	3	3	2	2

Repeated measures data in short and fat format

Here’s the data in the short and fat format. Look again at the scroll bar on the right. It is so large because we are seeing almost all of the data (37 out of the 40 rows). Notice also that the short and fat format fills the screen almost completely from left to right. The tall and thin format did not come anywhere close to the right margin.



A simple alternative to both tall/thin and short/fat

- Disadvantages of tall/thin
  - Too much repetition
- Disadvantages of short/fat
- Database format
  - Time constant table
  - Time varying table

There are serious disadvantages to the tall and thin format. For gender, you see male, male, male repeated twelve times because the first subject's gender was male for all twelve of the repeated measures. Repeated data like this is an open invitation for errors to creep in.

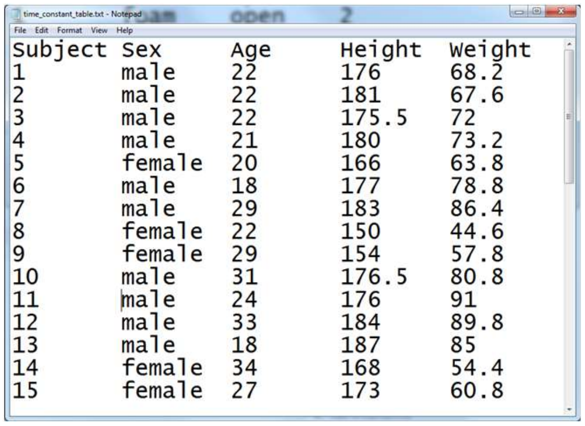
There are serious disadvantages to the short and fat format. Listing each visit in a separate set of columns or listing every repeated measurement condition in separate columns means that you have to do a lot of left and right scrolling to see all of the data. Left/right scrolling is painful, compared to up/down scrolling.

I had someone with a longitudinal data set where there were up to 40 separate visits with multiple measurements at each visit. He used the short and fat format and actually ran out of columns in the Excel spreadsheet. This was an earlier version of Excel where the column limit was 255. Now it is much higher. But think about it. A data set with 255 columns. Why I'd get exhausted just trying to name all those columns.

I recommend that you split longitudinal data into two pieces. The first piece represents time constant variables. These are the variables that do not change from visit to visit or across the repeated measurements. Most demographics like sex and race are the same from visit to visit.

The time varying table contains values that change from one visit to another in a longitudinal study or values that change from one repeated measures condition to another

Time constant data

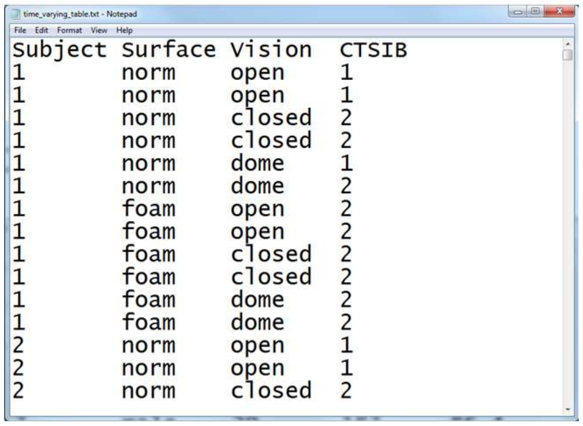


```
Subject Sex     Age     Height  Weight
1       male    22      176     68.2
2       male    22      181     67.6
3       male    22      175.5   72
4       male    21      180     73.2
5       female  20      166     63.8
6       male    18      177     78.8
7       male    29      183     86.4
8       female  22      150     44.6
9       female  29      154     57.8
10      male    31      176.5   80.8
11      male    24      176     91
12      male    33      184     89.8
13      male    18      187     85
14      female  34      168     54.4
15      female  27      173     60.8
```

Table listing time constant data only

There is one row per patient for the time constant variables and notice that there is no repetition.

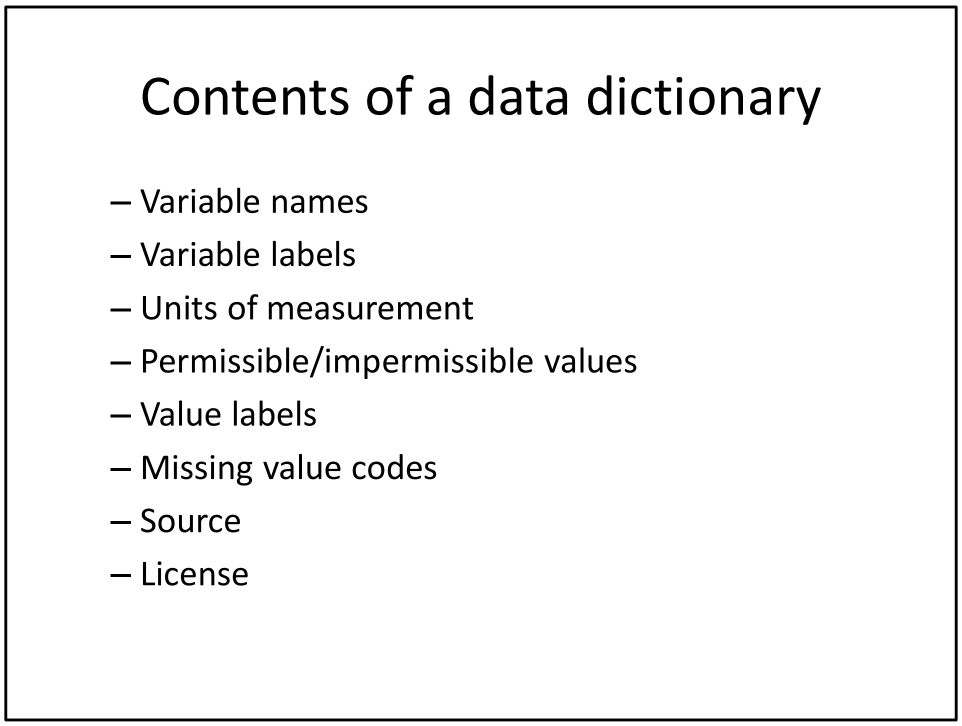
Time varying data


time_varying_table.txt - Notepad

```
Subject Surface Vision  CTSIB
1       norm    open    1
1       norm    open    1
1       norm    closed  2
1       norm    closed  2
1       norm    dome    1
1       norm    dome    2
1       foam    open    2
1       foam    open    2
1       foam    closed  2
1       foam    closed  2
1       foam    dome    2
1       foam    dome    2
2       norm    open    1
2       norm    open    1
2       norm    closed  2
```

Table listing time varying data only

There is one row per repeated measurement for the time varying variables. If you split the data this way, make sure that each patient is given a unique identifier. You'll need this when you merge the time constant and time varying variables.



Contents of a data dictionary

– Variable names
– Variable labels
– Units of measurement
– Permissible/impermissible values
– Value labels
– Missing value codes
– Source
– License

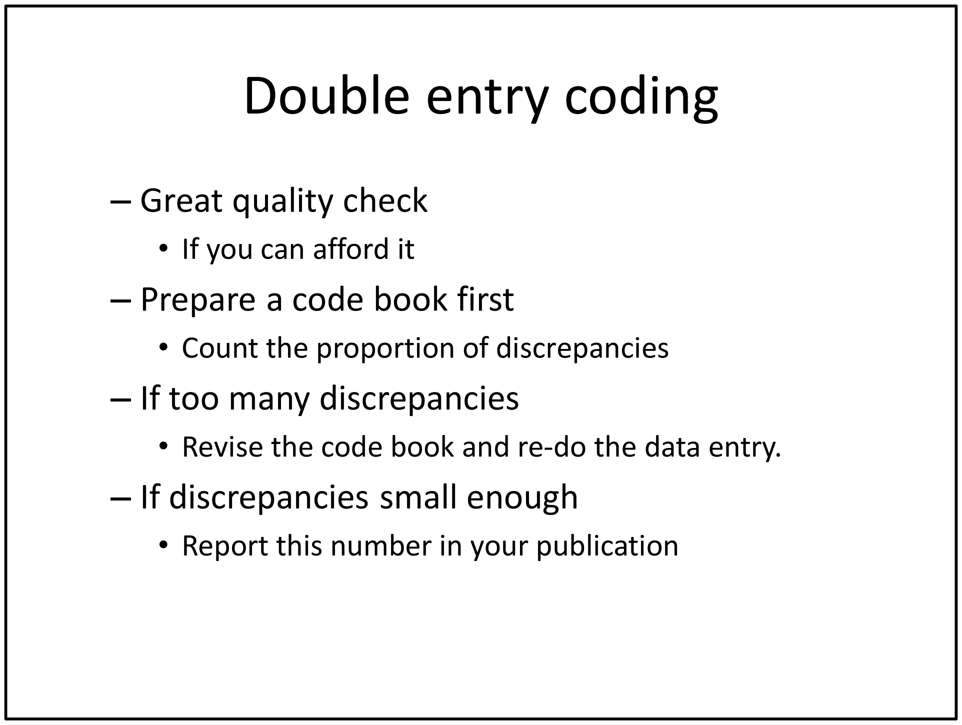
The data dictionary should list variable names. We already talked about variable names. Also include a longer description, a variable label. The variable label can use blanks, special symbols, and typically has a longer length. Be sure to specify the units of measurement, if applicable.

If there are permissible/impermissible values or a legal range, then include this information in the data dictionary. It could be put in as part of the variable label or listed as a footnote.

If your variable is categorical, specify what the categories are. There is nothing worse than not knowing whether 1 is male and 2 is female or if 1 is female and 2 is male.

Third break

- What have you learned
  - Multiple response variables
  - Longitudinal/repeated measures data
- What's next
  - Double entry coding
  - Excel files



Double entry coding

- Great quality check
  - If you can afford it
- Prepare a code book first
  - Count the proportion of discrepancies
- If too many discrepancies
  - Revise the code book and re-do the data entry.
- If discrepancies small enough
  - Report this number in your publication

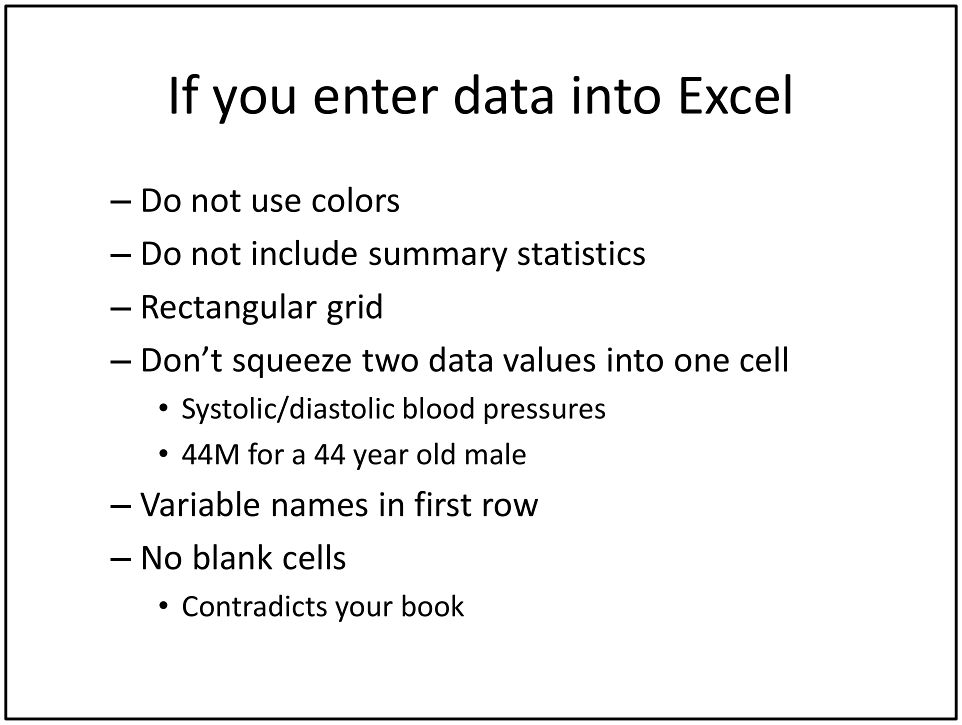
Double entry coding is using two people to enter the identical data sets. They do this independently, then you compare the results to measure the degree of data entry errors.

Before you start double entry coding, you need a very detailed code book so there is little room for subjective interpretations of what to enter for unusual responses.

Fix the discrepancies, of course. The error rate will be different for the different data entry people, but the product of the two error rates is an estimate of the number of undetected errors. So if one person had an error rate of 5% and the other had an error rate of 2%, the overall error rate would be 0.05*0.02=0.001.

If your undetected error rate is too high, you are at fault for writing an ambiguous code book. Rewrite the code book and repeat the whole process.

Once you get a good undetected error rate, publish this number in your methods or results section of your paper.



If you enter data into Excel

- – Do not use colors
- – Do not include summary statistics
- – Rectangular grid
- – Don’t squeeze two data values into one cell
  - Systolic/diastolic blood pressures
  - 44M for a 44 year old male
- – Variable names in first row
- – No blank cells
  - Contradicts your book

A lot of researchers do their data entry in Microsoft Excel or some similar spreadsheet program. This is, at best, sub-optimal. REDCap is a much better option. But if you decide to use a spreadsheet, be careful. Spreadsheets are very flexible. Too flexible. You can put some of your numbers over here and others over there and then some more in the middle, and what you end up with is a muddle.

The key rule for a spreadsheet is to use a rectangular grid. A rectangular grid has a single number at the intersection of every row and column. Don’t leave holes in your grid, even if the value is missing.

Don’t try to squeeze to pieces of data into the same cell. A common problem is when researchers put 120/80 into a cell to represent the systolic and diastolic blood pressures. When you enter data like this, you have two problems. First, there is no easy way to compute the average systolic or diastolic blood pressure. Second, some statistical software will see the slash and confuse it for division, converting your 120/80 into 1.5.

Use one cell for the systolic blood pressure and a different cell for the diastolic blood pressure. Likewise, an entry of 44M for a forty four year old male is troublesome. You can’t easily compute the average age or the percentage of males.

It’s a good idea to place the variable names in the first row of the spreadsheet. It’s not a strict requirement, as you have already provided the variable names in your data dictionary. If you are nice enough to do this, however, it will be easier to import the data. the first row also makes it possibile to spot check your data to make sure that the columns are lining up properly.

Don’t use the first two or three rows to organize your variables into headings and subheadings.

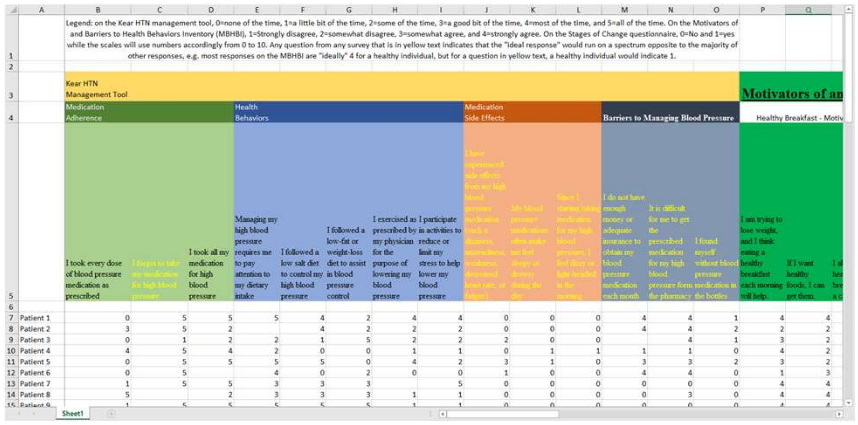
A lot of people will include some brief summary statistics as the last row or two of the spreadsheet. This is a bad idea, as the summary statistics might be confused for actual data points. If you need to compute a few summary statistics, learn how to place these in a separate worksheet.

Don’t use a lot of fancy formatting in your spreadsheet. This means, no colors, no different fonts or sizes, and no merged cells. All of this is lost when you import your data into SAS, SPSS, or R.

Let me emphasize one more thing. Do not use blank cells ever in your spreadsheet. If your data is missing, use a missing value code. This contradicts your book’s advice, but your book is wrong. A blank cell could easily be converted into a zero, if you’re not careful and zero is quite different than missing.

If the structure of your data is irregular and cannot fit into a rectangular grid, re-organize. If you have measurements on 24 treatment subjects and 26 controls, don’t put the outcome measure for the 24 treatment patients in one column and the 26 control patients in a second column. Put all 50 in a single column and add an additional column with 26 ones and 24 twos to distinguish who is a treatment patient and who is a control.

A poorly structured spreadsheet



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1		Legend: on the Kear HTN management tool, 0=none of the time, 1=a little bit of the time, 2=some of the time, 3=a good bit of the time, 4=most of the time, and 5=all of the time. On the Motivators of and Barriers to Health Behaviors Inventory (MBHBI), 1=Strongly disagree, 2=somewhat disagree, 3=somewhat agree, and 4=strongly agree. On the Stages of Change questionnaire, 0=No and 1=yes while the scales will use numbers accordingly from 0 to 10. Any question from any survey that is in yellow text indicates that the "ideal response" would run on a spectrum opposite to the majority of other responses, e.g. most responses on the MBHBI are "ideally" 4 for a healthy individual, but for a question in yellow text, a healthy individual would indicate 1.															
2																	
3		Kear HTN Management Tool														Motivators of an	
4		Medication Adherence			Health Behaviors					Medication Side Effects			Barriers to Managing Blood Pressure			Healthy Breakfast - Motiv	
5		I took every dose of blood pressure medication as prescribed	I forgot to take my medication for high blood pressure	I took all my medication for high blood pressure	Managing my high blood pressure requires me to pay attention to my dietary intake	I followed a low salt diet to control my high blood pressure	I followed a low-fat or weight-loss diet to assist in blood pressure control	I exercised as prescribed by my physician for the purpose of lowering my blood pressure	I participate in activities to reduce or limit my stress to help lower my blood pressure	I have experienced side effects from my high blood pressure medication (such a dizziness, unsteadiness, weakness, decreased heart rate, or fatigue)	My blood pressure medications often make me feel sleepy or drowsy during the day	Since I starting taking medication for my high blood pressure, I feel dizzy or light-headed in the morning	I do not have enough money or adequate insurance to obtain my blood pressure medication each month	It is difficult for me to get the prescribed medication for my high blood pressure form the pharmacy	I found myself without blood pressure medication in the bottles	I am trying to lose weight, and I think eating a healthy breakfast each morning will help	If I want healthy foods, I can get them
6																	
7	Patient 1	0	5	5	5	4	2	4	4	0	0	0	4	4	1	4	4
8	Patient 2	3	5	2		4	2	2	2	0	0	0	4	4	2	2	2
9	Patient 3	0	1	2	2	1	5	2	2	2	0	0		4	1	3	2
10	Patient 4	4	5	4	2	0	0	1	1	0	1	1	1	1	0	4	2
11	Patient 5	0	5	5	5	5	0	4	2	3	1	0	3	3	2	3	2
12	Patient 6	0	5		4	0	2	0	0	1	0	0	4	4	0	1	3
13	Patient 7	1	5	5	3	3	3		5	0	0	0	0	0	0	4	4
14	Patient 8	5		2	3	3	3	1	1	0	0	0	0	3	0	4	4
15	Patient 9	1	5	5	5	5	5	1	1	0	0	0	0	0	0	4	4

Sheet1

A spreadsheet with data

This image and the next three might be a bit tricky to see on your screen. You can download the original images from the Canvas site and magnify them, if you want to see the details.

Here is a spreadsheet that I received recently that violates many of the recommendations that I just made.

The first row contains information that should belong in the data dictionary. The second and third rows organize the variables into major groupings: the Kear HTN Management Tool, and the Motivators of and Barriers to Health-Smart Behaviors Inventory. There are subgroups within these two groups: Medication Adherence, Health Behaviors, Medication Side Effects, and Barriers to Managing Blood Pressure for the Kear HTN Management Tool, and Healthy Breakfast - Motivators, and some other subgrous for the Motivators of and Barriers to Health-Smart Behaviors Inventory.

The groupings and subgroupings belong in the data dictionary but not here.

The fourth column gives a description of each variable. The first two columns are: I forgot to take my medication for high blood pressure, I took all my medication for high blood pressure. These are much too long to serve as variable names. They would make great variable labels instead, and this information belongs in a data dictionary.

There is an important bit of information hidden in the formatting of the variable labels.

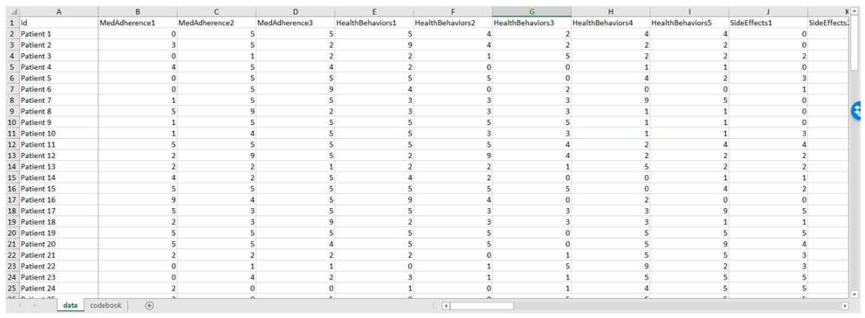
“Any question from any survey that is in yellow text indicates that the”ideal response” would run on a spectrum opposite to the majority of other responses, e.g. most responses on the MBHBI are “ideally” 4 for a healthy individual, but for a question in yellow text, a healthy individual would indicate 1.”

So the ninth column (I participate in activities to reduce or limit my stress to help lower my blood pressure) is in black because a large value is good. But the tenth column (I have experienced side effects from my high blood pressure medication (such a dizziness, unsteadiness, weakness, decreased heart rate, or fatigue)) is in yellow because a small value is good.

The sixth row is totally blank which is great for readability but an annoyance when you import the data.

Finally, notice the blank cells. I found out that these represented cases where someone skipped a question.

Revisions to this spreadsheet



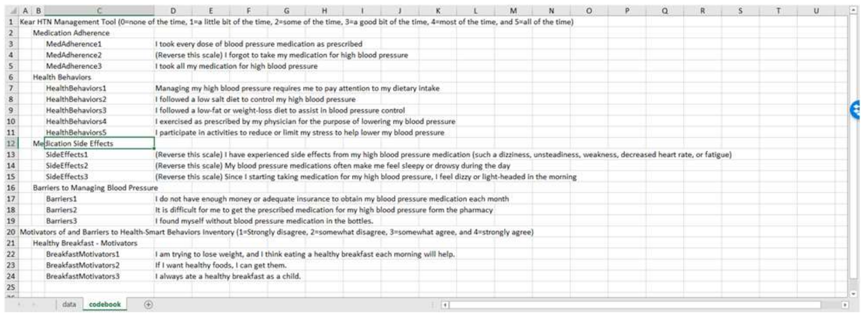
	A	B	C	D	E	F	G	H	I	J	K
1	id	MedAdherence1	MedAdherence2	MedAdherence3	HealthBehaviors1	HealthBehaviors2	HealthBehaviors3	HealthBehaviors4	HealthBehaviors5	SideEffects1	SideEffects
2	Patient 1	0	5	5	5	4	2	4	4	0	
3	Patient 2	3	5	2	9	4	2	2	2	0	
4	Patient 3	0	1	2	2	1	5	2	2	2	
5	Patient 4	4	5	4	2	0	0	1	1	0	
6	Patient 5	0	5	5	5	5	0	4	2	3	
7	Patient 6	0	5	9	4	0	2	0	0	1	
8	Patient 7	1	5	5	3	3	3	9	5	0	
9	Patient 8	5	9	2	3	3	3	1	1	0	
10	Patient 9	1	5	5	5	5	5	1	1	0	
11	Patient 10	1	4	5	5	3	3	1	1	3	
12	Patient 11	5	5	5	5	5	4	2	4	4	
13	Patient 12	2	9	5	2	9	4	2	2	2	
14	Patient 13	2	2	1	2	2	1	5	2	2	
15	Patient 14	4	2	5	4	2	0	0	1	1	
16	Patient 15	5	5	5	5	5	5	0	4	2	
17	Patient 16	9	4	5	9	4	0	2	0	0	
18	Patient 17	5	3	5	5	3	3	3	9	5	
19	Patient 18	2	3	9	2	3	3	3	1	1	
20	Patient 19	5	5	5	5	5	0	5	5	5	
21	Patient 20	5	5	4	5	5	0	5	9	4	
22	Patient 21	2	2	2	2	0	1	5	5	3	
23	Patient 22	0	1	1	0	1	5	9	2	3	
24	Patient 23	0	4	2	3	1	1	5	5	5	
25	Patient 24	2	0	0	1	0	1	4	5	5	

A revised and better organized spreadsheet

I was able to work with this data set, as is, without too much trouble, but here’s a modification that would have been a bit easier to work with.

The first row includes brief, but descriptive variable names. The column names are MedAdherence1 through 3, HealthBehaviors1 through 5, SideEffects1 through 3, and so forth. I anticipated that I would not need to distinguish among the particular questions within a subgroup, since I would probably just use an aggregate sum of each subgroup (after reverse scaling) for most of the analyses.

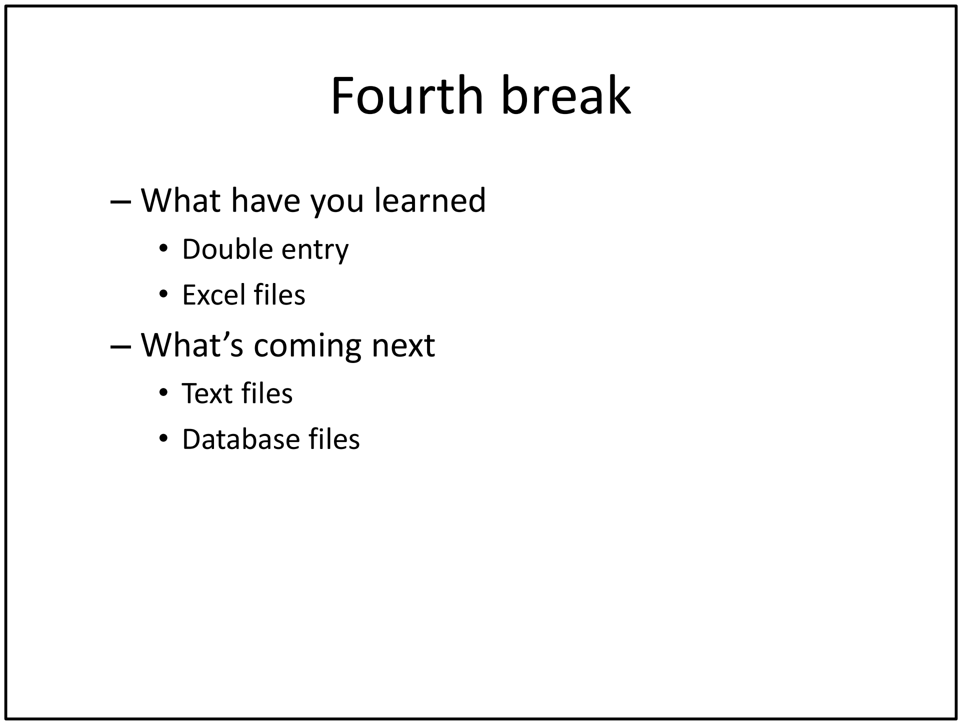
The codebook from this spreadsheet



	A	B	C	D
1	Kear HTN Management Tool (0=none of the time, 1=a little bit of the time, 2=some of the time, 3=a good bit of the time, 4=most of the time, and 5=all of the time)			
2		Medication Adherence		
3			MedAdherence1	I took every dose of blood pressure medication as prescribed
4			MedAdherence2	(Reverse this scale) I forgot to take my medication for high blood pressure
5			MedAdherence3	I took all my medication for high blood pressure
6		Health Behaviors		
7			HealthBehaviors1	Managing my high blood pressure requires me to pay attention to my dietary intake
8			HealthBehaviors2	I followed a low salt diet to control my high blood pressure
9			HealthBehaviors3	I followed a low-fat or weight-loss diet to assist in blood pressure control
10			HealthBehaviors4	I exercised as prescribed by my physician for the purpose of lowering my blood pressure
11			HealthBehaviors5	I participate in activities to reduce or limit my stress to help lower my blood pressure
12		Medication Side Effects		
13			SideEffects1	(Reverse this scale) I have experienced side effects from my high blood pressure medication (such a dizziness, unsteadiness, weakness, decreased heart rate, or fatigue)
14			SideEffects2	(Reverse this scale) My blood pressure medications often make me feel sleepy or drowsy during the day
15			SideEffects3	(Reverse this scale) Since I starting taking medication for my high blood pressure, I feel dizzy or light-headed in the morning
16		Barriers to Managing Blood Pressure		
17			Barriers1	I do not have enough money or adequate insurance to obtain my blood pressure medication each month
18			Barriers2	It is difficult for me to get the prescribed medication for my high blood pressure form the pharmacy
19			Barriers3	I found myself without blood pressure medication in the bottles.
20	Motivators of and Barriers to Health-Smart Behaviors Inventory (1=Strongly disagree, 2=somewhat disagree, 3=somewhat agree, and 4=strongly agree)			
21		Healthy Breakfast - Motivators		
22			BreakfastMotivators1	I am trying to lose weight, and I think eating a healthy breakfast each morning will help.
23			BreakfastMotivators2	If I want healthy foods, I can get them.
24			BreakfastMotivators3	I always ate a healthy breakfast as a child.
25				

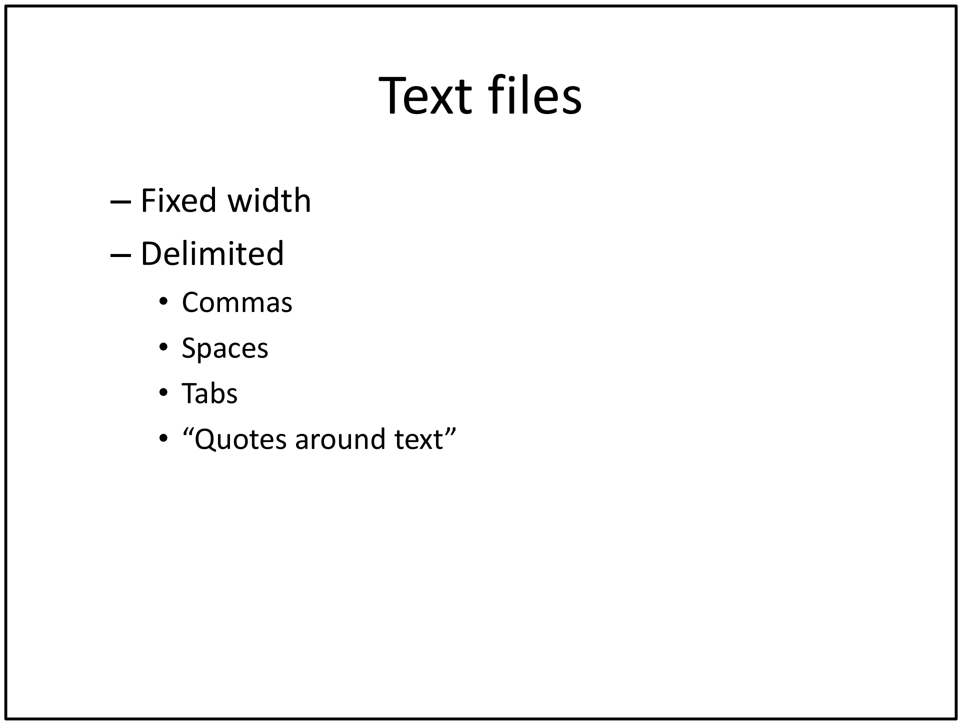
A codebook associated with revised spreadsheet

I also cut and pasted some of the preliminary rows in the original spreadsheet and restructured things a bit to develop a data dictionary.



Fourth break

- What have you learned
  - Double entry
  - Excel files
- What's coming next
  - Text files
  - Database files



Text files

- – Fixed width
- – Delimited
  - Commas
  - Spaces
  - Tabs
  - “Quotes around text”

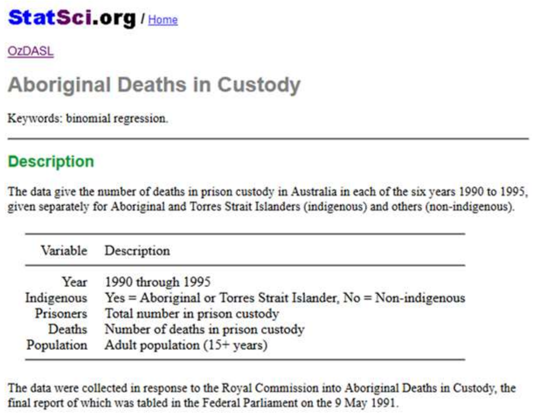
You may wish to create a text file for others to use. The text format is popular for sharing your data, because others can inspect the data easily without needing any specialized software.

You can used a fixed width format. The individual data values are limited to specific columns and are not allowed to creep over into the wrong columns. It takes a bit of planning to develop a fixed width format, because you need to specify, in advance, how many columns are enough to fit any possible value for your data. You may know this for some data. An Apgar score, for example, would require two columns, while an IQ score would require thee columns. But how many columns would you need for a person’s name?

A delimited file uses a special character, called a delimiter, to separate individual data values within a row. The common delimiters are commas, spaces, and tabs. Often delimited files will place quote marks around text, especially for comma delimited files.

I want to show some examples of each of these types of text files.

Data dictionary for aboriginal prison death study



StatSci.org / Home

OzDASL

Aboriginal Deaths in Custody

Keywords: binomial regression.

Description

The data give the number of deaths in prison custody in Australia in each of the six years 1990 to 1995, given separately for Aboriginal and Torres Strait Islanders (indigenous) and others (non-indigenous).

Variable	Description
Year	1990 through 1995
Indigenous	Yes = Aboriginal or Torres Strait Islander, No = Non-indigenous
Prisoners	Total number in prison custody
Deaths	Number of deaths in prison custody
Population	Adult population (15+ years)

The data were collected in response to the Royal Commission into Aboriginal Deaths in Custody, the final report of which was tabled in the Federal Parliament on the 9 May 1991.

Data dictionary

Here is the data dictionary for a small data set that I will use to illustrate the various text formats. It comes from OzDASL, the Australian Data And Story Library. This data set represents the number of deaths in prison of both aboriginal Australians and others. The variables are

Year (1990 through 1995),

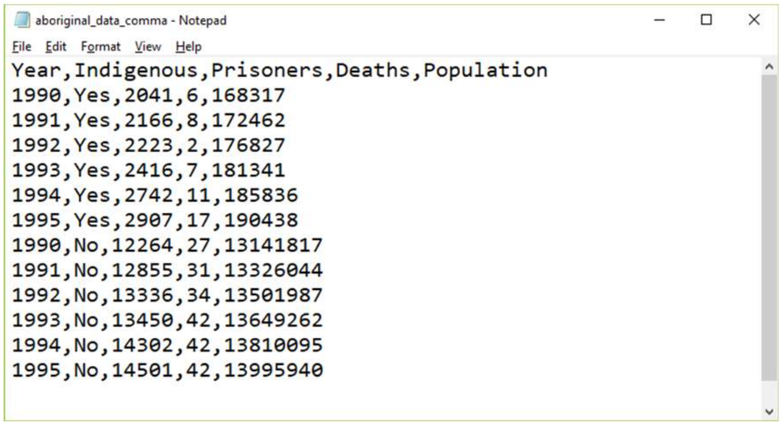
Indigenous (yes or no),

Prisoners, the number in prison custody,

Deaths, the number of deaths, and

Population, the adult population (15 years or older)

Comma separated values (csv)

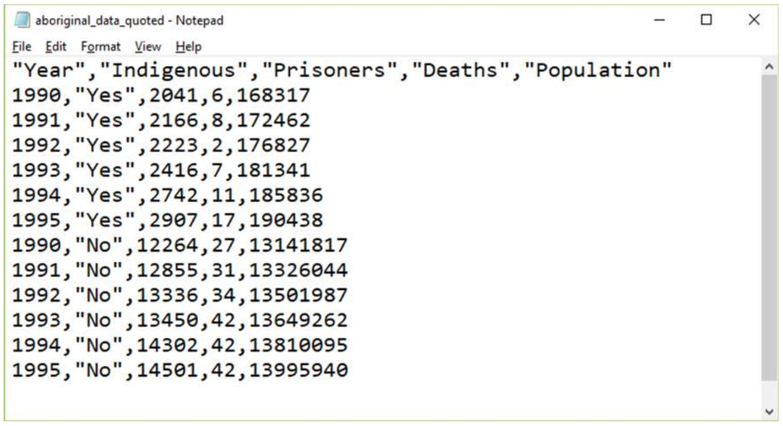


```
Year,Indigenous,Prisoners,Deaths,Population
1990,Yes,2041,6,168317
1991,Yes,2166,8,172462
1992,Yes,2223,2,176827
1993,Yes,2416,7,181341
1994,Yes,2742,11,185836
1995,Yes,2907,17,190438
1990,No,12264,27,13141817
1991,No,12855,31,13326044
1992,No,13336,34,13501987
1993,No,13450,42,13649262
1994,No,14302,42,13810095
1995,No,14501,42,13995940
```

Data set using a comma separated value format

The most popular text format is the comma separated value. These files usually have a .csv extension. Each data point is separated from the other data values by a comma.

Comma separated values with quotes


aboriginal_data_quoted - Notepad

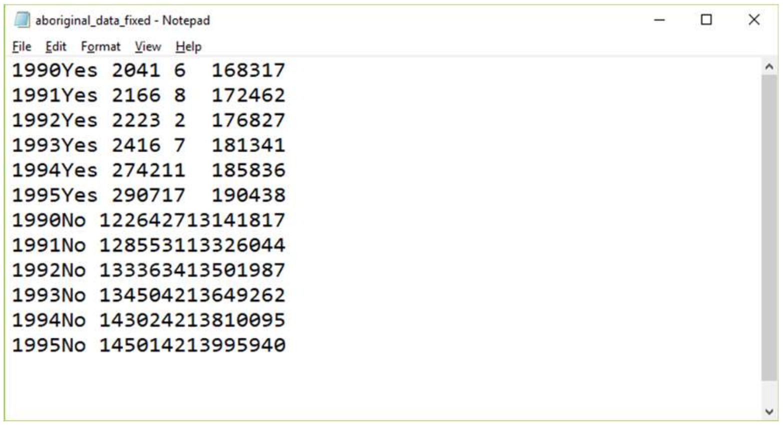
```
"Year","Indigenous","Prisoners","Deaths","Population"
1990,"Yes",2041,6,168317
1991,"Yes",2166,8,172462
1992,"Yes",2223,2,176827
1993,"Yes",2416,7,181341
1994,"Yes",2742,11,185836
1995,"Yes",2907,17,190438
1990,"No",12264,27,13141817
1991,"No",12855,31,13326044
1992,"No",13336,34,13501987
1993,"No",13450,42,13649262
1994,"No",14302,42,13810095
1995,"No",14501,42,13995940
```

Data set using a quoted format

If your text data contains space, punctuation, or other special symbols, then it is a good idea to place quotes around your text. It's not really needed for this data set, but I wanted to show you anyway.

There are elaborate rules for how to handle text data that contains quote marks. Use double quotes to surround text that includes single quotes and use single quotes to surround text that has double quotes. Doubling up a quote mark is another approach that usually works. It's one of those tedious things that only Statisticians worry about.

Fixed width format



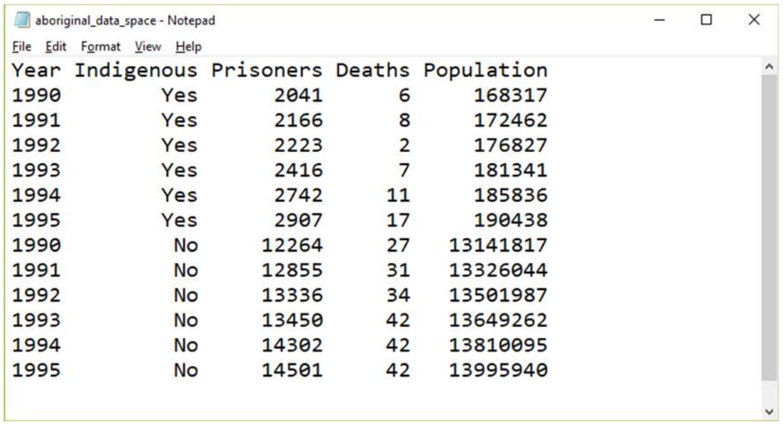
```
1990Yes 2041 6  168317
1991Yes 2166 8  172462
1992Yes 2223 2  176827
1993Yes 2416 7  181341
1994Yes 274211  185836
1995Yes 290717  190438
1990No 12264271314181 7
1991No 128553113326044
1992No 133363413501987
1993No 134504213649262
1994No 143024213810095
1995No 145014213995940
```

Data set using a fixed width format

In this fixed width format, the first variable, year, is listed in columns 1 through 4. Then the text variable, Indigenous, takes the next three columns, columns 5 through 7. The number of prisoners is either a four or five digit number, so it uses columns 8 through 12. The number of deaths is, thankfully small, either one or two digits, so it fits into columns 13 and 14. The population could be in the tens of millions, so it needs to use columns 15 through 22.

You need a fairly detailed data dictionary for fixed with format data, because otherwise you might not be sure where one number stops and another number begins.

Spaced format

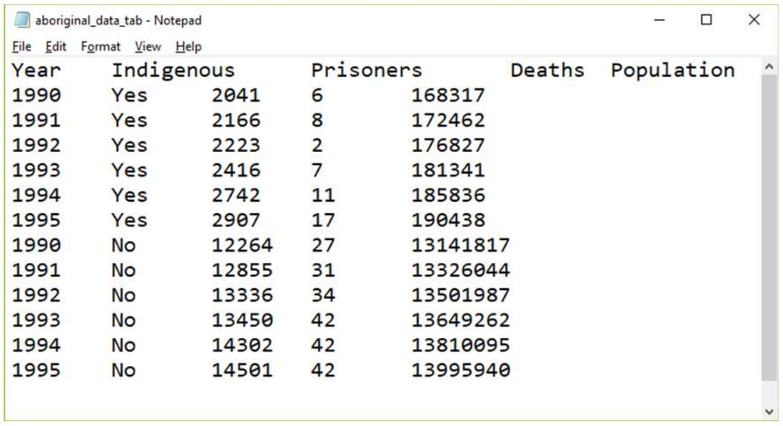


```
Year Indigenous Prisoners Deaths Population
1990        Yes      2041      6     168317
1991        Yes      2166      8     172462
1992        Yes      2223      2     176827
1993        Yes      2416      7     181341
1994        Yes      2742     11     185836
1995        Yes      2907     17     190438
1990         No     12264     27   13141817
1991         No     12855     31   13326044
1992         No     13336     34   13501987
1993         No     13450     42   13649262
1994         No     14302     42   13810095
1995         No     14501     42   13995940
```

Data set using a spaced format

Usually, you will encounter a fixed width format in data sets with hundreds of variables and thousands or millions of rows. In these files, saving a few characters here or there can make a big difference in download times. But for small data sets, you can insert a few extra blanks so that everything lines up nicely. Here is an example.

Tab separated values


aboriginal_data_tab - Notepad

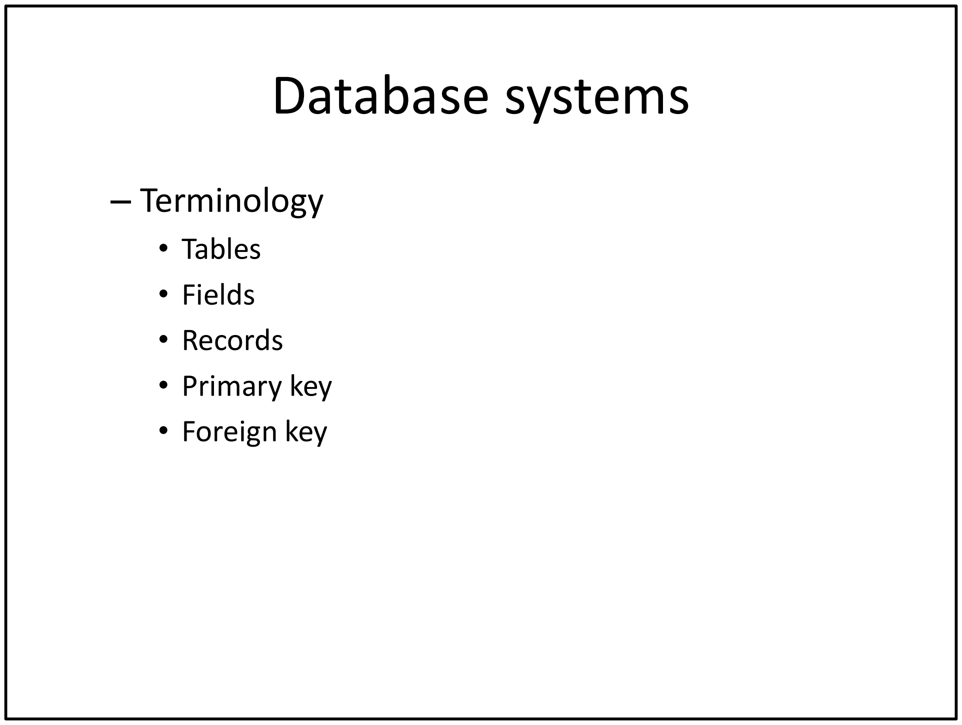
```
Year    Indigenous      Prisoners       Deaths  Population
1990    Yes     2041    6       168317
1991    Yes     2166    8       172462
1992    Yes     2223    2       176827
1993    Yes     2416    7       181341
1994    Yes     2742    11      185836
1995    Yes     2907    17      190438
1990    No      12264   27      13141817
1991    No      12855   31      13326044
1992    No      13336   34      13501987
1993    No      13450   42      13649262
1994    No      14302   42      13810095
1995    No      14501   42      13995940
```

Data set using a tab separated value format

In the tab separated format, the tab character is used to separate values. The tab character effectively jumps to a specific fixed location. These locations are usually eight characters apart, starting at column 9, then 17, 25, etc.

The tab character appears to the human eye as multiple blanks but it is actually a non printing character, 09 in hexadecimal.

It's hard sometimes to figure out if a file is using tabs or spaces as delimiters. Most text editors and word processors will let you search for the tab character, often with a code like /t (forward slash t). If you start to insert blanks in a tab delimited file, at first nothing happens, then all of a sudden, you notice a sudden jump in the text to the right of where you are inserting. Finally, look to see if the text seems to line up somewhat but with a few irregularities. This happens when some of the text data for a variable is longer than 8 characters and other text for that variable is shorter than 8 characters. In the example shown here, none of the data is longer than 8 characters, but the variable names are, which is why they don't align perfectly with the data.

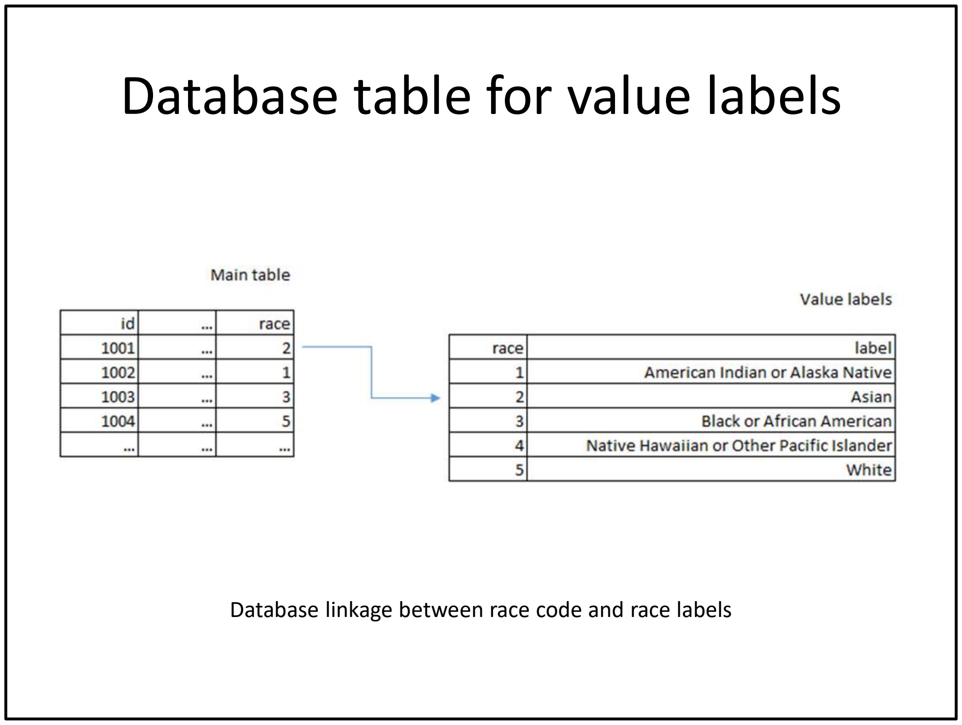


Database systems

– Terminology
  - Tables
  - Fields
  - Records
  - Primary key
  - Foreign key

Databases use slightly different terminology than what statistical software like R, SAS, and SPSS use, so let's review the basic terminology.

A table is a rectangular grid of numbers. Fields represent the columns of the table, what our statistical packages would call variables. Records represents the rows of the table, or what our statistical pacakges would call observations. The primary key is a identifier that, ideally, is unique for every record. A foreign key is a link between this table and a the primary key of a different table.



Database table for value labels

Main table

id	...	race
1001	...	2
1002	...	1
1003	...	3
1004	...	5
...	...	...

Value labels

race	label
1	American Indian or Alaska Native
2	Asian
3	Black or African American
4	Native Hawaiian or Other Pacific Islander
5	White

Database linkage between race code and race labels

Earlier, you saw how longitudinal or repeated measures data could be split into two tables. The use of multiple linked tables is a powerful reason to use a database.

Here's another example. Suppose you were tracking race using the following categories:

American Indian or Alaska Native

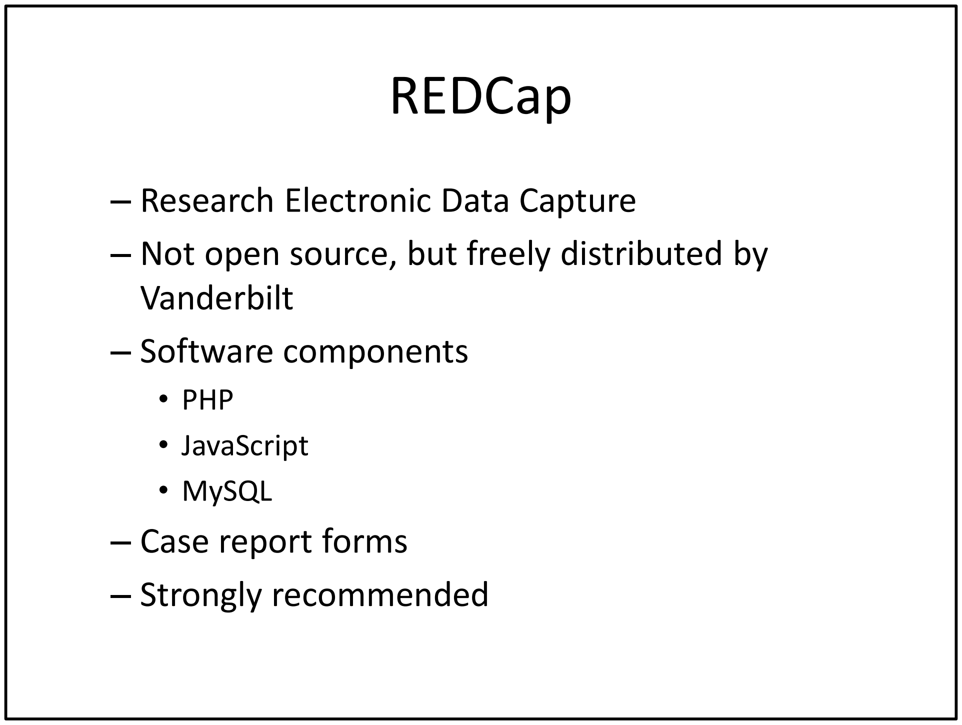
Asian

Black or African American

Native Hawaiian or Other Pacific Islander

White

You wouldn't want to type the long labels in over and over again. So you would use the numbers 1 through 5 with 1 for American Indian or Alaska Native and 5 for White. But in a separate table you could keep the longer labels and then just join the tables together to get the nice labels for each subject.



REDCap

– Research Electronic Data Capture
– Not open source, but freely distributed by Vanderbilt
– Software components
  • PHP
  • JavaScript
  • MySQL
– Case report forms
– Strongly recommended

REDCap is a database built on top of open source components, PHP, JavaScript, and MySQL. It has a lot of features that are especially attractive to researchers. For data entry, it allows you to easily define a case report form, a structure for your data entry. It can manage double entry seamlessly. When you are done with data entry, it has nice features to simplify the export of your data into R, SAS, or SPSS.

REDCap also has a survey module that allows you to collect questionnaire data over the Internet.

It has a lot of nice built in security features as well to guard against accidental release of private health information.

You should use REDCap for pretty much any research project. It has a bit less value, perhaps, for secondary data sets, but for any project where you have to enter the data yourself, you'd be crazy not to use REDCap.

Conclusion

- Data dictionary
  - Variable names, Variable labels, Value labels, Missing value codes
- Managing complex files
  - Multiple response, Longitudinal/repeated measures data
- Storage options
  - Spreadsheet, Text file, Database, REDCap