ETC1010: Data Modelling and Computing

Lecture 2: Tidy data

Di Cook (dicook@monash.edu, @visnut)

Week 2

Overview

- Terminology of statistical data exploration
- Process of exploring data
- Making basic plots
- Different forms of data and making them tidy

Exploratory data analysis

"Data science is an exciting discipline that allows you to turn raw data into understanding, insight, and knowledge." Grolemund and Wickham

Exploratory data analysis is the stage in the analysis where the analyst "plays in the sand" with their data to see what it has to tell them. It allows us to find the unexpected, and come to some understanding of the information that the data contains. Cook and Swayne

Exploring your data, stages

- import
- tidy
- transform/wrangle/clean
- visualise
- model
- communicate

Rectangular data

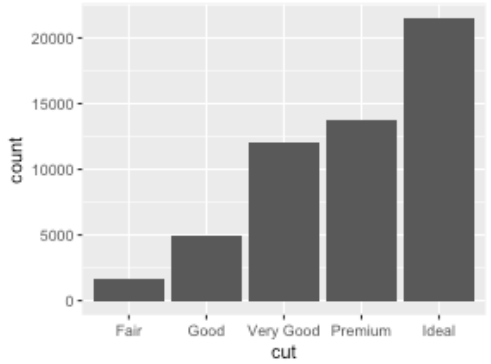
- Variables are in the columns
  - A **variable** is a quantity, quality, or property that you can measure.
  - A value is the state of a variable when you measure it. The value of a variable may change from measurement to measurement.
- Observations are in the rows: An **observation** is a set of measurements made under similar conditions (you usually make all of the measurements in an observation at the same time and on the same object). An observation will contain several values, each associated with a different variable. I'll sometimes refer to an observation as a data point.
- **Tabular data** is a set of values, each associated with a variable and an observation. Tabular data is **tidy** if each value is placed in its own "cell", each variable in its own column, and each observation in its own row.

Fundamental questions

- What type of variation occurs within my variables?
- What type of covariation occurs between my variables?

Visualising distributions

```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut))
```



`Cut` is a categorical variable, show we display it using a bar chart. Height of the bar shows the number in each category. What is the most common diamond cut?

Your turn

In the previous plot:

- What is the variable?
- What are the observations?

Under the hood

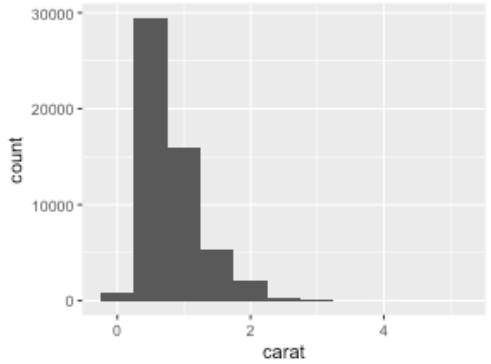
The counts for the bar chart were automatically calculated, prior to plotting:

```
diamonds %>%
  count(cut)
# A tibble: 5 x 2
        cut     n
      <ord> <int>
1      Fair  1610
2      Good  4906
3 Very Good 12082
4   Premium 13791
5     Ideal 21551
```

Visualising distributions

A variable is continuous if it can take any of an infinite set of ordered values. To examine the distribution of a continuous variable, use a histogram:

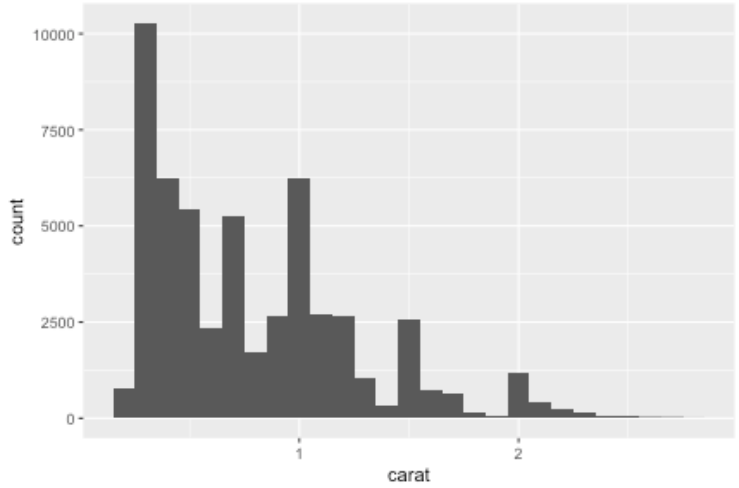
```
ggplot(data = diamonds) +
  geom_histogram(mapping = aes(x = carat), binwidth = 0.5)
```



Carat ranges from 0 to 5 centered at 0.5, and has a right-skewed shape. This means that most diamonds are around 0.5-1 carat, and there are very few large diamonds.

Focus

Because most of the information about carats of most diamonds is actually in the smaller range, zoom in to this domain to see detail.



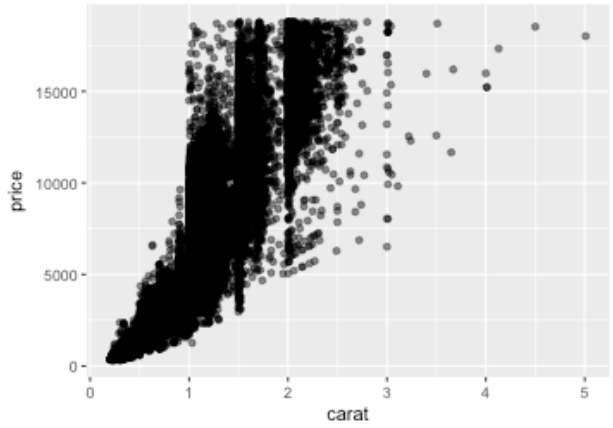
Your turn

The distribution for diamonds smaller than 3 carats is multi-modal, at this 0.1 binwidth resolution. There are peaks at 0.5, 1, 1.5 and 2 carats. What might this mean?

Covariation

If we examine price of the diamond relative to carat, we might expect the price increases with carat. A scatterplot is a way to explore this covariation.

```
ggplot(diamonds, aes(x=carat, y=price)) + geom_point(alpha=0.5)
```

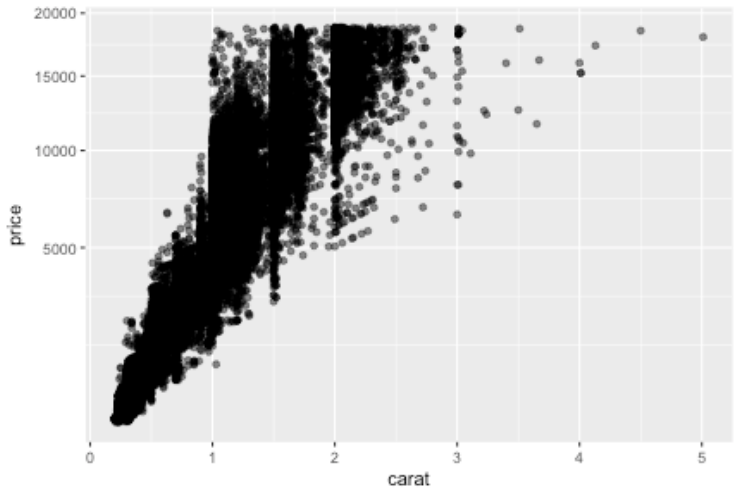


There is a positive association between the two variables, but it is also nonlinear, and there are a few outliers, very large and not so expensive diamonds. We can also see concentrations of points at fixed values, 1, 1.5, 2, ... The variation is not uniform for all values of carat, as carat increases so does the variation. This is called heteroskedastic.

Linearise

Nonlinear relationships are harder to model, so at this point it can be helpful to transform one or more of the variables. Here we have put price on a square root scale, which makes the relationship more linear.

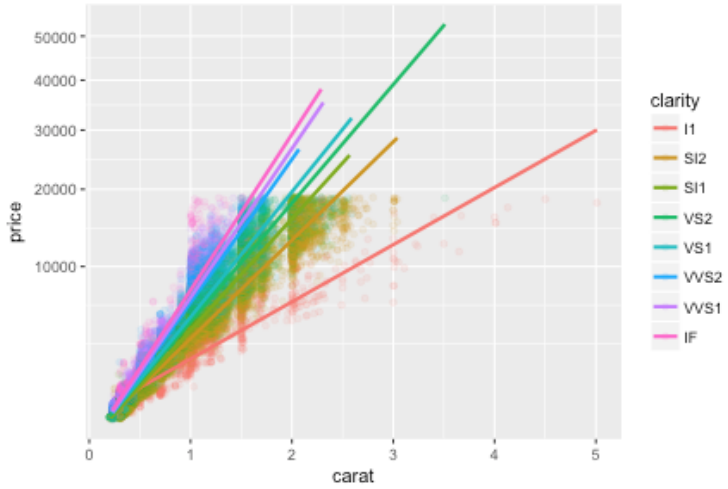
```
ggplot(diamonds, aes(x=carat, y=price)) + geom_point(alpha=0.5) + scale_y_sqr
```



Covariation

The reason that the larger diamonds don't bring relatively larger prices might be related to quality. Incorporating this into the plot may help tease the variation apart.

```
ggplot(diamonds, aes(x=carat, y=price, colour=clarity)) +
  geom_point(alpha=0.1) +
  geom_smooth(se=FALSE, method="lm") + scale_y_sqrt()
```



Tidying data

Here are a bunch of examples data sets that have come across my desk. We are going to work out what are the variables and what are the observations, to know what form we need to rearrange the data to get it into tidy form.

Example 1

What are the variables? Observations?

```
# A tibble: 6 x 4
                      Inst AvNumPubs AvNumCits PctCompletion
                     <chr>     <dbl>     <dbl>          <dbl>
1 ARIZONA STATE UNIVERSITY      0.90      1.57           31.7
2        AUBURN UNIVERSITY      0.79      0.64           44.4
3           BOSTON COLLEGE      0.51      1.03           46.8
4        BOSTON UNIVERSITY      0.49      2.66           34.2
5      BRANDEIS UNIVERSITY      0.30      3.03           48.7
6         BROWN UNIVERSITY      0.84      2.31           54.6
```

Is it in tidy form?

Example 2

What are the variables? Observations?

```
# A tibble: 3 x 12
      id `WI-6.R1` `WI-6.R2` `WI-6.R4` `WM-6.R1` `WM-6.R2` `WI-12.R1`
   <chr>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>      <dbl>
1 Gene 1  2.182424 2.2042219  4.195636 2.6273345  5.063641   4.540002
2 Gene 2  1.464224 0.5854472  1.859238 0.5152242  2.882808   1.364037
3 Gene 3  2.031792 0.8701078  3.281983 0.5330452  4.627315   2.182192
# ... with 5 more variables: `WI-12.R2` <dbl>, `WI-12.R4` <dbl>,
#   `WM-12.R1` <dbl>, `WM-12.R2` <dbl>, `WM-12.R4` <dbl>
```

Example 3

What are the variables? Observations?

```
           V1   V2 V3   V4  V5  V9 V13 V17 V21 V25 V29 V33 V37 V41 V45 V49
1 ASN00086282 1970  7 TMAX 141 124 113 123 148 149 139 153 123 108 119 112
2 ASN00086282 1970  7 TMIN  80  63  36  57  69  47  84  78  49  42  48  56
3 ASN00086282 1970  7 PRCP   3  30   0   0  36   3   0   0  10  23   3   0
4 ASN00086282 1970  8 TMAX 145 128 150 122 109 112 116 142 166 127 117 127
5 ASN00086282 1970  8 TMIN  50  61  75  67  41  51  48  -7  56  62  47  33
6 ASN00086282 1970  8 PRCP   0  66   0  53  13   3   8   0   0   0   3   5
  V53 V57 V61 V65 V69 V73 V77 V81 V85 V89 V93 V97
1 126 112 115 133 134 126 104 143 141 134 117 142
2  51  36  44  39  40  58  15  33  51  74  39  66
3   5   0   0   0   0   0   8   0  18   0   0   0
4 159 143 114  65 113 125 129 147 161 168 178 161
5  67  84  11  41  18  50  22  28  74  94  73  88
6   0   0  64   3  99  36   8   0   0   0   8  36
```

Example 4

What are the variables? Observations?

```
# A tibble: 6 x 22
   iso2  year  m_04 m_514 m_014 m_1524 m_2534 m_3544 m_4554 m_5564  m_65
  <chr> <int> <int> <int> <int>  <int>  <int>  <int>  <int>  <int> <int>
1    ZW  2003    NA    NA   133    874   3048   2228    981    367   205
2    ZW  2004    NA    NA   187    833   2908   2298   1056    366   198
3    ZW  2005    NA    NA   210    837   2264   1855    762    295   656
4    ZW  2006    NA    NA   215    736   2391   1939    896    348   199
5    ZW  2007     6   132   138    500   3693      0    716    292   153
6    ZW  2008    NA    NA   127    614      0   3316    704    263   185
# ... with 11 more variables: m_u <int>, f_04 <int>, f_514 <int>,
#   f_014 <int>, f_1524 <int>, f_2534 <int>, f_3544 <int>, f_4554 <int>,
#   f_5564 <int>, f_65 <int>, f_u <int>
```

Example 5

	religion	<$10k	$10-20k	$20-30k	$30-40k
1	Agnostic	27	34	60	81
2	Atheist	12	27	37	52
3	Buddhist	27	21	30	34
4	Catholic	418	617	732	670
5	Don’t know/refused	15	14	15	11

Example 6

10 week sensory experiment, 12 individuals assessed taste of french fries on several scales (how potato-y, buttery, grassy, rancid, paint-y do they taste?), fried in one of 3 different oils, replicated twice. First few rows:

	time	treatment	subject	rep	potato	buttery	grassy	rancid	painty
61	1	1	3	1	2.9	0.0	0.0	0.0	5.5
25	1	1	3	2	14.0	0.0	0.0	1.1	0.0
62	1	1	10	1	11.0	6.4	0.0	0.0	0.0
26	1	1	10	2	9.9	5.9	2.9	2.2	0.0

What is the experimental unit? What are the factors of the experiment? What was measured? What do you want to know?

Messy Data Patterns

There are various features of messy data that one can observe in practice. Here are some of the more commonly observed patterns:

- Column headers are values, not variable names
- Variables are stored in both rows and columns, contingency table format
- One type of experimental unit stored in multiple tables
- Dates in many different formats

What is Tidy Data?

- Each observation forms a row
- Each variable forms a column
- Data is contained in a single table
- Long form makes it easier to reshape in many different ways
- Wide form is common for analysis


Description by Hadley Wickham
Tidy data = lego
http://www.flickr.com/photos/wwworks/2473052504


Description by Hadley Wickham
Messy data = play mobile
https://www.flickr.com/photos/kafka4prez/57282282

Tidy Verbs

- `gather`: specify the **keys** (identifiers) and the **values** (measures) to make long form (used to be called melting)
- `spread`: variables in columns (used to be called casting)
- `nest`/`unnest`: working with list variables
- `separate`/`unite`: split and combine columns

Tidying the example 2 data

```
genes <- read_csv("data/genes.csv")
head(genes)
# A tibble: 3 x 12
      id `WI-6.R1` `WI-6.R2` `WI-6.R4` `WM-6.R1` `WM-6.R2` `WI-12.R1`
   <chr>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>      <dbl>
1 Gene 1  2.182424 2.2042219  4.195636 2.6273345  5.063641   4.540002
2 Gene 2  1.464224 0.5854472  1.859238 0.5152242  2.882808   1.364037
3 Gene 3  2.031792 0.8701078  3.281983 0.5330452  4.627315   2.182192
# ... with 5 more variables: `WI-12.R2` <dbl>, `WI-12.R4` <dbl>,
#   `WM-12.R1` <dbl>, `WM-12.R2` <dbl>, `WM-12.R4` <dbl>
```

Gather Column Names into Long Form

```
gather(genes, variable, expr, -id)
# A tibble: 33 x 3
       id variable      expr
    <chr>    <chr>     <dbl>
 1 Gene 1  WI-6.R1 2.1824242
 2 Gene 2  WI-6.R1 1.4642236
 3 Gene 3  WI-6.R1 2.0317925
 4 Gene 1  WI-6.R2 2.2042219
 5 Gene 2  WI-6.R2 0.5854472
 6 Gene 3  WI-6.R2 0.8701078
 7 Gene 1  WI-6.R4 4.1956364
 8 Gene 2  WI-6.R4 1.8592378
 9 Gene 3  WI-6.R4 3.2819835
10 Gene 1  WM-6.R1 2.6273345
# ... with 23 more rows
```

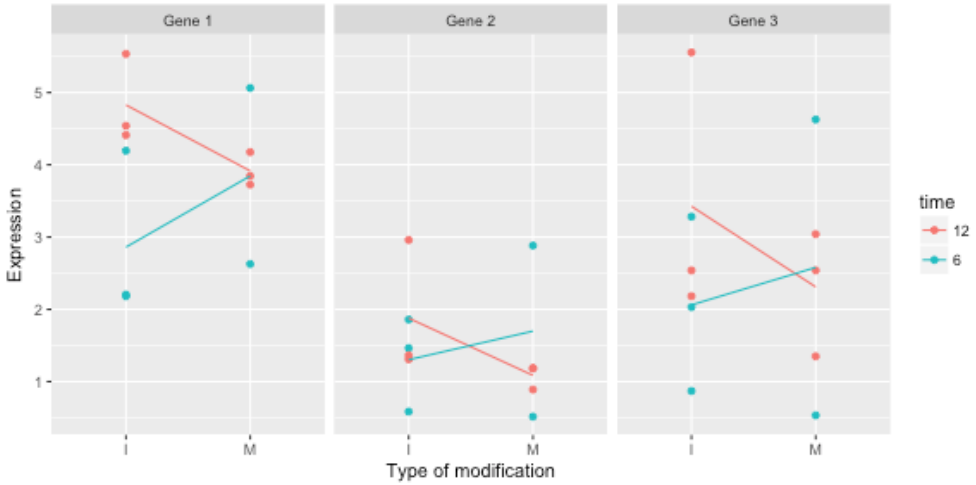
Separate Columns

```
genes %>%
  gather(variable, expr, -id) %>%
  separate(variable, c("trt", "leftover"), "-")
# A tibble: 33 x 4
       id   trt leftover      expr
 *  <chr> <chr>    <chr>     <dbl>
 1 Gene 1    WI     6.R1 2.1824242
 2 Gene 2    WI     6.R1 1.4642236
 3 Gene 3    WI     6.R1 2.0317925
 4 Gene 1    WI     6.R2 2.2042219
 5 Gene 2    WI     6.R2 0.5854472
 6 Gene 3    WI     6.R2 0.8701078
 7 Gene 1    WI     6.R4 4.1956364
 8 Gene 2    WI     6.R4 1.8592378
 9 Gene 3    WI     6.R4 3.2819835
10 Gene 1    WM     6.R1 2.6273345
# ... with 23 more rows
```

```
genes %>%
  gather(variable, expr, -id) %>%
  separate(variable, c("trt", "leftover"), "-") %>%
  separate(leftover, c("time", "rep"), "\\.")
# A tibble: 33 x 5
       id   trt  time   rep      expr
 *  <chr> <chr> <chr> <chr>     <dbl>
 1 Gene 1    WI     6    R1 2.1824242
 2 Gene 2    WI     6    R1 1.4642236
 3 Gene 3    WI     6    R1 2.0317925
 4 Gene 1    WI     6    R2 2.2042219
 5 Gene 2    WI     6    R2 0.5854472
 6 Gene 3    WI     6    R2 0.8701078
 7 Gene 1    WI     6    R4 4.1956364
 8 Gene 2    WI     6    R4 1.8592378
 9 Gene 3    WI     6    R4 3.2819835
10 Gene 1    WM     6    R1 2.6273345
# ... with 23 more rows
```

```
gtidy <- genes %>%
  gather(variable, expr, -id) %>%
  separate(variable, c("trt", "leftover"), "-") %>%
  separate(leftover, c("time", "rep"), "\\.") %>%
  mutate(trt = sub("W", "", trt)) %>%
  mutate(rep = sub("R", "", rep))
head(gtidy)
# A tibble: 6 x 5
      id   trt  time   rep      expr
   <chr> <chr> <chr> <chr>     <dbl>
1 Gene 1     I     6     1 2.1824242
2 Gene 2     I     6     1 1.4642236
3 Gene 3     I     6     1 2.0317925
4 Gene 1     I     6     2 2.2042219
5 Gene 2     I     6     2 0.5854472
6 Gene 3     I     6     2 0.8701078
```

Make a picture



Tidying example 3

```
melbtemp <- read.fwf("data/ASN00086282.dly",
                     c(11, 4, 2, 4, rep(c(5, 1, 1, 1), 31)), fill=T)
melbtemp <- melbtemp[,c(1,2,3,4,seq(5,128,4))]
colnames(melbtemp) <- c("id", "year", "month", "var", paste0("V",1:31))
head(melbtemp)
           id year month  var  V1  V2  V3  V4  V5  V6  V7  V8  V9 V10 V11
1 ASN00086282 1970     7 TMAX 141 124 113 123 148 149 139 153 123 108 119
2 ASN00086282 1970     7 TMIN  80  63  36  57  69  47  84  78  49  42  48
3 ASN00086282 1970     7 PRCP   3  30   0   0  36   3   0   0  10  23   3
4 ASN00086282 1970     8 TMAX 145 128 150 122 109 112 116 142 166 127 117
5 ASN00086282 1970     8 TMIN  50  61  75  67  41  51  48  -7  56  62  47
6 ASN00086282 1970     8 PRCP   0  66   0  53  13   3   8   0   0   0   3
  V12 V13 V14 V15 V16 V17 V18 V19 V20 V21 V22 V23 V24 V25 V26 V27 V28 V29
1 112 126 112 115 133 134 126 104 143 141 134 117 142 158 149 133 143 150
2  56  51  36  44  39  40  58  15  33  51  74  39  66  78  36  61  46  42
3   0   5   0   0   0   0   0   8   0  18   0   0   0   0  13   3   0  25
4 127 159 143 114  65 113 125 129 147 161 168 178 161 145 142 137 150 120
5  33  67  84  11  41  18  50  22  28  74  94  73  88  50  48  54  78  47
6   5   0   0  64   3  99  36   8   0   0   0   8  36  25  30  56   5  69
  V30 V31
1 145 115
2  63  39
3   0   3
4 114 129
5  18  39
6   3  20
```

```
melbtemp %>%
  gather(day, value, V1:V31) %>%
  head()
          id year month  var day value
1 ASN00086282 1970     7 TMAX  V1   141
2 ASN00086282 1970     7 TMIN  V1    80
3 ASN00086282 1970     7 PRCP  V1     3
4 ASN00086282 1970     8 TMAX  V1   145
5 ASN00086282 1970     8 TMIN  V1    50
6 ASN00086282 1970     8 PRCP  V1     0
```

```
melbtemp %>%
  gather(day, value, V1:V31) %>%
  mutate(day = sub("V", "", day)) %>%
  head()
           id year month  var day value
1 ASN00086282 1970     7 TMAX   1   141
2 ASN00086282 1970     7 TMIN   1    80
3 ASN00086282 1970     7 PRCP   1     3
4 ASN00086282 1970     8 TMAX   1   145
5 ASN00086282 1970     8 TMIN   1    50
6 ASN00086282 1970     8 PRCP   1     0
```

Missing values have been coded as -9999. Need to recode these as NA.

```
    min median  max
1 -9999     91 1388
```

```
melbtemp %>%
  gather(day, value, V1:V31) %>%
  mutate(day = sub("V", "", day)) %>%
  mutate(value=ifelse(value==-9999, NA, value)) %>%
  head()
          id year month  var day value
1 ASN00086282 1970     7 TMAX   1   141
2 ASN00086282 1970     7 TMIN   1    80
3 ASN00086282 1970     7 PRCP   1     3
4 ASN00086282 1970     8 TMAX   1   145
5 ASN00086282 1970     8 TMIN   1    50
6 ASN00086282 1970     8 PRCP   1     0
```

There are more variables types that are recognisable:

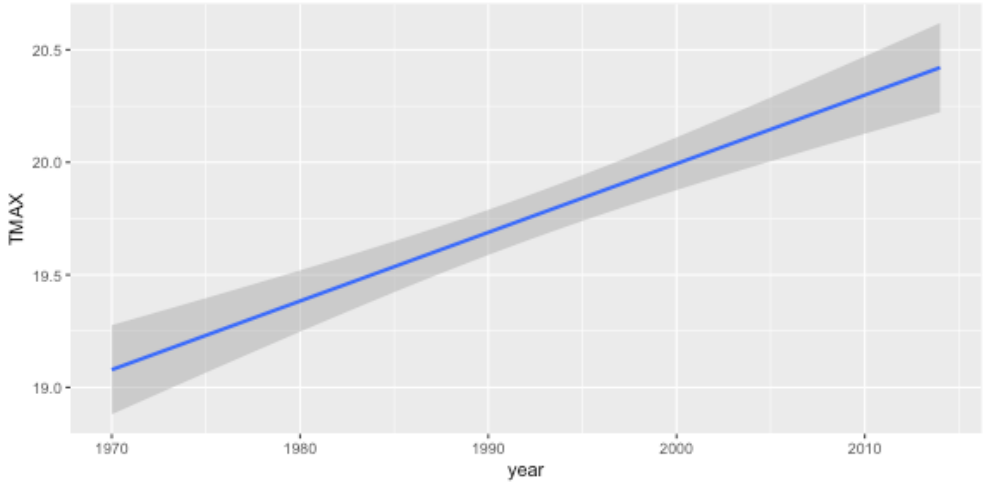
```
# A tibble: 7 x 2
     var     n
  <fctr> <int>
1   DATN    62
2   DATX    31
3   MDTN    62
4   MDTX    31
5   PRCP 16399
6   TMAX 16399
7   TMIN 16399
```

```
melbtemp %>%
  gather(day, value, V1:V31) %>%
  mutate(day = sub("V", "", day)) %>%
  mutate(value=ifelse(value==-9999, NA, value)) %>%
  filter(var %in% c("PRCP", "TMAX", "TMIN")) %>%
  head()
          id year month  var day value
1 ASN00086282 1970     7 TMAX   1   141
2 ASN00086282 1970     7 TMIN   1    80
3 ASN00086282 1970     7 PRCP   1     3
4 ASN00086282 1970     8 TMAX   1   145
5 ASN00086282 1970     8 TMIN   1    50
6 ASN00086282 1970     8 PRCP   1     0
```

```
melbtemp %>%
  gather(day, value, V1:V31) %>%
  mutate(day = sub("V", "", day)) %>%
  mutate(value=ifelse(value==-9999, NA, value)) %>%
  filter(var %in% c("PRCP", "TMAX", "TMIN")) %>%
  spread(var, value) %>%
  head()
          id year month day PRCP TMAX TMIN
1 ASN00086282 1970     7   1    3  141   80
2 ASN00086282 1970     7  10   23  108   42
3 ASN00086282 1970     7  11    3  119   48
4 ASN00086282 1970     7  12    0  112   56
5 ASN00086282 1970     7  13    5  126   51
6 ASN00086282 1970     7  14    0  112   36
```

```
melbtemp %>%
  gather(day, value, V1:V31) %>%
  mutate(day = sub("V", "", day)) %>%
  mutate(value=ifelse(value==-9999, NA, value)) %>%
  filter(var %in% c("PRCP", "TMAX", "TMIN")) %>%
  spread(var, value) %>%
  mutate(PRCP=PRCP/10, TMAX=TMAX/10, TMIN=TMIN/10) %>%
  head()
          id year month day PRCP TMAX TMIN
1 ASN00086282 1970     7   1  0.3 14.1  8.0
2 ASN00086282 1970     7  10  2.3 10.8  4.2
3 ASN00086282 1970     7  11  0.3 11.9  4.8
4 ASN00086282 1970     7  12  0.0 11.2  5.6
5 ASN00086282 1970     7  13  0.5 12.6  5.1
6 ASN00086282 1970     7  14  0.0 11.2  3.6
```

Now look at Melbourne max temperatures



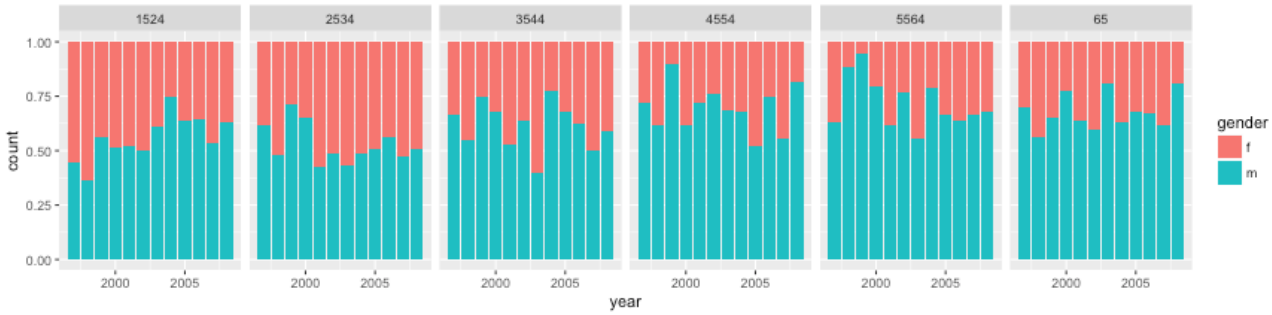
Tidying example 4

```
tb <- tb %>%
  gather(var, count, m_04:f_u) %>%
  separate(var, c("gender", "age")) %>%
  filter(!(age %in% c("014", "04", "514", "u")))
head(tb)
```

```
# A tibble: 6 x 5
   iso2  year gender   age count
  <chr> <int>  <chr> <chr> <int>
1    AD  1989      m  1524    NA
2    AD  1990      m  1524    NA
3    AD  1991      m  1524    NA
4    AD  1992      m  1524    NA
5    AD  1993      m  1524    NA
6    AD  1994      m  1524    NA
```

Numbers in Australia

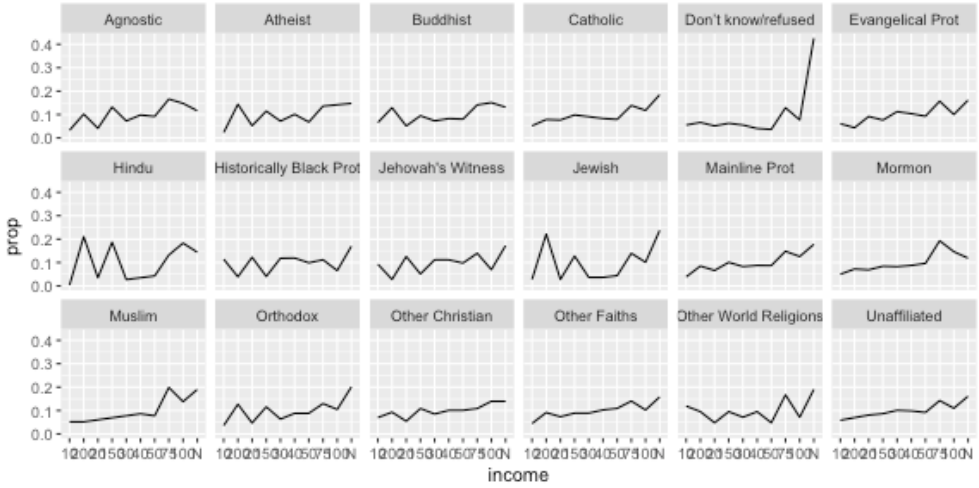
```
tb %>%
  filter(iso2 == "AU") %>%
  ggplot(aes(x = year, y = count, fill = gender)) +
  geom_bar(stat = "identity", position = "fill") +
  facet_grid(~ age)
```



Tidying example 5

```
pew %>%
  gather(income, count, -religion) %>%
  group_by(religion) %>%
  mutate(prop = count / sum(count)) %>%
  head()
# A tibble: 6 x 4
# Groups:   religion [6]
             religion income count       prop
                <chr>  <chr> <int>      <dbl>
1           Agnostic  <$10k    27 0.03268765
2            Atheist  <$10k    12 0.02330097
3           Buddhist  <$10k    27 0.06569343
4           Catholic  <$10k   418 0.05189968
5 Don't know/refused  <$10k    15 0.05514706
6   Evangelical Prot  <$10k   575 0.06070524
```

Relationship between income and religion



Share and share alike