Introduction to R Packages for Data Management

Dawn Koffman
Office of Population Research (OPR)
Princeton University
May 9, 2016

Diagram shown here based on David Robinson, 4/11/2015, “broom: an R Package to Convert Statistical Models into Tidy Data Frames”
Tidy Data

Tidy data sets provide a **standardized** way to link physical layout of a data set with its meaning.

Data sets come in many formats ...but many R tools work best with one format, a tidy data set.

Source: RStudio Data Wrangling Cheatsheet
RStudio Data Visualization Cheatsheet
Tidy Format

Some terms:
- a data set is a collection of values
- many data sets are organized as rectangular tables made up of rows and columns

Values are organized in two ways:

- every value belongs to a variable and an observation
  - a variable contains all values that measure the same underlying attribute
    (for example, life expectancy or total fertility rate) across units
  - an observation contains all values measured on the same unit
    (for example, country-year)

Each **variable** is saved in its own column and column headers are variable names, not values.
Each **observation** is saved in its own row.
Each “type” of observation is stored in a single table.

Principles of tidy data are very similar to principles of relational database design ...
although terminology is a bit different.
## Data Set Example 1

**Tidy?**

<table>
<thead>
<tr>
<th>year</th>
<th>le</th>
<th>le_male</th>
<th>le_female</th>
<th>le_w</th>
<th>le_wmale</th>
<th>le_wfemale</th>
<th>le_b</th>
<th>le_bmale</th>
<th>le_bfemale</th>
</tr>
</thead>
<tbody>
<tr>
<td>1900</td>
<td>47.3</td>
<td>46.3</td>
<td>48.3</td>
<td>47.6</td>
<td>46.6</td>
<td>48.7</td>
<td>33.0</td>
<td>32.5</td>
<td>33.5</td>
</tr>
<tr>
<td>1901</td>
<td>49.1</td>
<td>47.6</td>
<td>50.6</td>
<td>49.4</td>
<td>48.0</td>
<td>51.0</td>
<td>33.7</td>
<td>32.2</td>
<td>35.3</td>
</tr>
<tr>
<td>1902</td>
<td>51.5</td>
<td>49.8</td>
<td>53.4</td>
<td>51.9</td>
<td>50.2</td>
<td>53.8</td>
<td>34.6</td>
<td>32.9</td>
<td>36.4</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1994</td>
<td>75.7</td>
<td>72.4</td>
<td>79.0</td>
<td>76.5</td>
<td>73.3</td>
<td>79.6</td>
<td>69.5</td>
<td>64.9</td>
<td>73.9</td>
</tr>
<tr>
<td>1995</td>
<td>75.8</td>
<td>72.5</td>
<td>78.9</td>
<td>76.5</td>
<td>73.4</td>
<td>79.6</td>
<td>69.6</td>
<td>65.2</td>
<td>73.9</td>
</tr>
<tr>
<td>1996</td>
<td>76.1</td>
<td>73.1</td>
<td>79.1</td>
<td>76.8</td>
<td>73.9</td>
<td>79.7</td>
<td>70.2</td>
<td>66.1</td>
<td>74.2</td>
</tr>
<tr>
<td>1997</td>
<td>76.5</td>
<td>73.6</td>
<td>79.4</td>
<td>77.2</td>
<td>74.3</td>
<td>79.9</td>
<td>71.1</td>
<td>67.2</td>
<td>74.7</td>
</tr>
<tr>
<td>1998</td>
<td>76.7</td>
<td>73.8</td>
<td>79.5</td>
<td>77.3</td>
<td>74.5</td>
<td>80.0</td>
<td>71.3</td>
<td>67.6</td>
<td>74.8</td>
</tr>
<tr>
<td>1999</td>
<td>76.7</td>
<td>73.9</td>
<td>79.4</td>
<td>77.3</td>
<td>74.6</td>
<td>79.9</td>
<td>71.4</td>
<td>67.8</td>
<td>74.7</td>
</tr>
</tbody>
</table>
# Data Set Example 1

Tidy?

<table>
<thead>
<tr>
<th>year</th>
<th>le</th>
<th>le_male</th>
<th>le_female</th>
<th>le_w</th>
<th>le_wmale</th>
<th>le_wfemale</th>
<th>le_b</th>
<th>le_bmale</th>
<th>le_bfemale</th>
</tr>
</thead>
<tbody>
<tr>
<td>1900</td>
<td>47.3</td>
<td>46.3</td>
<td>48.3</td>
<td>47.6</td>
<td>46.6</td>
<td>48.7</td>
<td>33.0</td>
<td>32.5</td>
<td>33.5</td>
</tr>
<tr>
<td>1901</td>
<td>49.1</td>
<td>47.6</td>
<td>50.6</td>
<td>49.4</td>
<td>48.0</td>
<td>51.0</td>
<td>33.7</td>
<td>32.2</td>
<td>35.3</td>
</tr>
<tr>
<td>1902</td>
<td>51.5</td>
<td>49.8</td>
<td>53.4</td>
<td>51.9</td>
<td>50.2</td>
<td>53.8</td>
<td>34.6</td>
<td>32.9</td>
<td>36.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1994</td>
<td>75.7</td>
<td>72.4</td>
<td>79.0</td>
<td>76.5</td>
<td>73.3</td>
<td>79.6</td>
<td>69.5</td>
<td>64.9</td>
<td>73.9</td>
</tr>
<tr>
<td>1995</td>
<td>75.8</td>
<td>72.5</td>
<td>78.9</td>
<td>76.5</td>
<td>73.4</td>
<td>79.6</td>
<td>69.6</td>
<td>65.2</td>
<td>73.9</td>
</tr>
<tr>
<td>1996</td>
<td>76.1</td>
<td>73.1</td>
<td>79.1</td>
<td>76.8</td>
<td>73.9</td>
<td>79.7</td>
<td>70.2</td>
<td>66.1</td>
<td>74.2</td>
</tr>
<tr>
<td>1997</td>
<td>76.5</td>
<td>73.6</td>
<td>79.4</td>
<td>77.2</td>
<td>74.3</td>
<td>79.9</td>
<td>71.1</td>
<td>67.2</td>
<td>74.7</td>
</tr>
<tr>
<td>1998</td>
<td>76.7</td>
<td>73.8</td>
<td>79.5</td>
<td>77.3</td>
<td>74.5</td>
<td>80.0</td>
<td>71.3</td>
<td>67.6</td>
<td>74.8</td>
</tr>
<tr>
<td>1999</td>
<td>76.7</td>
<td>73.9</td>
<td>79.4</td>
<td>77.3</td>
<td>74.6</td>
<td>79.9</td>
<td>71.4</td>
<td>67.8</td>
<td>74.7</td>
</tr>
</tbody>
</table>

Column headers contain **values**: male, female, b, w
## Data Set Example 2

### Tidy?

<table>
<thead>
<tr>
<th>country</th>
<th>continent</th>
<th>year</th>
<th>lifeExp</th>
<th>pop</th>
<th>gdpPercap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>1992</td>
<td>41.674</td>
<td>16317921</td>
<td>649.3414</td>
</tr>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>1997</td>
<td>41.763</td>
<td>22227425</td>
<td>635.3414</td>
</tr>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>2002</td>
<td>42.129</td>
<td>25268405</td>
<td>726.7341</td>
</tr>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>2007</td>
<td>43.828</td>
<td>31889923</td>
<td>974.5803</td>
</tr>
<tr>
<td>Albania</td>
<td>Europe</td>
<td>1992</td>
<td>71.581</td>
<td>3326498</td>
<td>2497.4379</td>
</tr>
<tr>
<td>Albania</td>
<td>Europe</td>
<td>1997</td>
<td>72.950</td>
<td>3428038</td>
<td>3193.0546</td>
</tr>
<tr>
<td>Albania</td>
<td>Europe</td>
<td>2002</td>
<td>75.651</td>
<td>3508512</td>
<td>4604.2117</td>
</tr>
<tr>
<td>Albania</td>
<td>Europe</td>
<td>2007</td>
<td>76.423</td>
<td>3600523</td>
<td>5937.0295</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>Zambia</td>
<td>Africa</td>
<td>1992</td>
<td>46.100</td>
<td>8381163</td>
<td>1210.8846</td>
</tr>
<tr>
<td>Zambia</td>
<td>Africa</td>
<td>1997</td>
<td>40.238</td>
<td>9417789</td>
<td>1071.3538</td>
</tr>
<tr>
<td>Zambia</td>
<td>Africa</td>
<td>2002</td>
<td>39.193</td>
<td>10595811</td>
<td>1071.6139</td>
</tr>
<tr>
<td>Zambia</td>
<td>Africa</td>
<td>2007</td>
<td>42.384</td>
<td>11746035</td>
<td>1271.2116</td>
</tr>
<tr>
<td>Zimbabwe</td>
<td>Africa</td>
<td>1992</td>
<td>60.377</td>
<td>10704340</td>
<td>693.4208</td>
</tr>
<tr>
<td>Zimbabwe</td>
<td>Africa</td>
<td>1997</td>
<td>46.809</td>
<td>11404948</td>
<td>792.4500</td>
</tr>
<tr>
<td>Zimbabwe</td>
<td>Africa</td>
<td>2002</td>
<td>39.989</td>
<td>11926563</td>
<td>672.0386</td>
</tr>
<tr>
<td>Zimbabwe</td>
<td>Africa</td>
<td>2007</td>
<td>43.487</td>
<td>12311143</td>
<td>469.7093</td>
</tr>
</tbody>
</table>
**Data Set Example 2**

**Tidy?**

<table>
<thead>
<tr>
<th>country</th>
<th>continent</th>
<th>year</th>
<th>lifeExp</th>
<th>pop</th>
<th>gdpPercap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>1992</td>
<td>41.674</td>
<td>16317921</td>
<td>649.3414</td>
</tr>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>1997</td>
<td>41.763</td>
<td>22227415</td>
<td>635.3414</td>
</tr>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>2002</td>
<td>42.129</td>
<td>25268405</td>
<td>726.7341</td>
</tr>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>2007</td>
<td>43.828</td>
<td>31889923</td>
<td>974.5803</td>
</tr>
<tr>
<td>Albania</td>
<td>Europe</td>
<td>1992</td>
<td>71.581</td>
<td>3326498</td>
<td>2497.4379</td>
</tr>
<tr>
<td>Albania</td>
<td>Europe</td>
<td>1997</td>
<td>72.950</td>
<td>3428038</td>
<td>3193.0546</td>
</tr>
<tr>
<td>Albania</td>
<td>Europe</td>
<td>2002</td>
<td>75.651</td>
<td>3508512</td>
<td>4604.2117</td>
</tr>
<tr>
<td>Albania</td>
<td>Europe</td>
<td>2007</td>
<td>76.423</td>
<td>3600523</td>
<td>5937.0295</td>
</tr>
<tr>
<td>Zambia</td>
<td>Africa</td>
<td>1992</td>
<td>46.100</td>
<td>8381163</td>
<td>1210.8846</td>
</tr>
<tr>
<td>Zambia</td>
<td>Africa</td>
<td>1997</td>
<td>40.238</td>
<td>9417789</td>
<td>1071.3538</td>
</tr>
<tr>
<td>Zambia</td>
<td>Africa</td>
<td>2002</td>
<td>39.193</td>
<td>10595811</td>
<td>1071.6139</td>
</tr>
<tr>
<td>Zambia</td>
<td>Africa</td>
<td>2007</td>
<td>42.384</td>
<td>11746035</td>
<td>1271.2116</td>
</tr>
<tr>
<td>Zimbabwe</td>
<td>Africa</td>
<td>1992</td>
<td>60.377</td>
<td>10704340</td>
<td>693.4208</td>
</tr>
<tr>
<td>Zimbabwe</td>
<td>Africa</td>
<td>1997</td>
<td>46.809</td>
<td>11404948</td>
<td>792.4500</td>
</tr>
<tr>
<td>Zimbabwe</td>
<td>Africa</td>
<td>2002</td>
<td>39.989</td>
<td>11926563</td>
<td>672.0386</td>
</tr>
<tr>
<td>Zimbabwe</td>
<td>Africa</td>
<td>2007</td>
<td>43.487</td>
<td>12311143</td>
<td>469.7093</td>
</tr>
</tbody>
</table>

Data about two different types of observations are stored in the same table:
- continent is an attribute of country
- lifeExp, pop and gdpPercap are attributes of country-year

Storing variable continent in a table where unit of observation is country-year is redundant and therefore wastes space and is prone to error.
### Data Set Example 3

**Tidy?**

<table>
<thead>
<tr>
<th>country</th>
<th>pop2012</th>
<th>imr</th>
<th>tfr</th>
<th>le</th>
<th>leM</th>
<th>leF</th>
<th>region</th>
<th>area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algeria</td>
<td>37.4</td>
<td>24</td>
<td>2.9</td>
<td>73</td>
<td>72</td>
<td>75</td>
<td>Northern Africa</td>
<td>Africa</td>
</tr>
<tr>
<td>Egypt</td>
<td>82.3</td>
<td>24</td>
<td>2.9</td>
<td>72</td>
<td>70</td>
<td>74</td>
<td>Northern Africa</td>
<td>Africa</td>
</tr>
<tr>
<td>Libya</td>
<td>6.5</td>
<td>14</td>
<td>2.6</td>
<td>75</td>
<td>72</td>
<td>77</td>
<td>Northern Africa</td>
<td>Africa</td>
</tr>
<tr>
<td>Morocco</td>
<td>32.6</td>
<td>30</td>
<td>2.3</td>
<td>72</td>
<td>70</td>
<td>74</td>
<td>Northern Africa</td>
<td>Africa</td>
</tr>
<tr>
<td>Sth Sudan</td>
<td>9.4</td>
<td>101</td>
<td>5.4</td>
<td>52</td>
<td>50</td>
<td>53</td>
<td>Northern Africa</td>
<td>Africa</td>
</tr>
<tr>
<td>Sudan</td>
<td>33.5</td>
<td>67</td>
<td>4.2</td>
<td>60</td>
<td>58</td>
<td>62</td>
<td>Northern Africa</td>
<td>Africa</td>
</tr>
<tr>
<td>Tunisia</td>
<td>10.8</td>
<td>20</td>
<td>2.1</td>
<td>75</td>
<td>73</td>
<td>77</td>
<td>Northern Africa</td>
<td>Africa</td>
</tr>
<tr>
<td>Benin</td>
<td>9.4</td>
<td>81</td>
<td>5.4</td>
<td>56</td>
<td>54</td>
<td>58</td>
<td>Western Africa</td>
<td>Africa</td>
</tr>
<tr>
<td>Gambia</td>
<td>1.8</td>
<td>70</td>
<td>4.9</td>
<td>58</td>
<td>57</td>
<td>59</td>
<td>Western Africa</td>
<td>Africa</td>
</tr>
<tr>
<td>Ghana</td>
<td>25.5</td>
<td>47</td>
<td>4.2</td>
<td>64</td>
<td>63</td>
<td>65</td>
<td>Western Africa</td>
<td>Africa</td>
</tr>
<tr>
<td>Serbia</td>
<td>7.1</td>
<td>7</td>
<td>1.3</td>
<td>74</td>
<td>71</td>
<td>77</td>
<td>Southern Europe</td>
<td>Europe</td>
</tr>
<tr>
<td>Slovenia</td>
<td>2.1</td>
<td>3</td>
<td>1.5</td>
<td>80</td>
<td>76</td>
<td>83</td>
<td>Southern Europe</td>
<td>Europe</td>
</tr>
<tr>
<td>Spain</td>
<td>46.2</td>
<td>3</td>
<td>1.4</td>
<td>82</td>
<td>79</td>
<td>85</td>
<td>Southern Europe</td>
<td>Europe</td>
</tr>
<tr>
<td>Australia</td>
<td>22.0</td>
<td>4</td>
<td>1.9</td>
<td>82</td>
<td>80</td>
<td>84</td>
<td>Oceania</td>
<td>Oceania</td>
</tr>
<tr>
<td>New Zealand</td>
<td>4.4</td>
<td>5</td>
<td>2.1</td>
<td>81</td>
<td>79</td>
<td>83</td>
<td>Oceania</td>
<td>Oceania</td>
</tr>
</tbody>
</table>
**Data Set Example 3**

**Tidy?**

<table>
<thead>
<tr>
<th>country</th>
<th>pop2012</th>
<th>imr</th>
<th>tfr</th>
<th>le</th>
<th>leM</th>
<th>leF</th>
<th>region</th>
<th>area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algeria</td>
<td>37.4</td>
<td>24</td>
<td>2.9</td>
<td>73</td>
<td>72</td>
<td>75</td>
<td>Northern Africa</td>
<td>Africa</td>
</tr>
<tr>
<td>Egypt</td>
<td>82.3</td>
<td>24</td>
<td>2.9</td>
<td>72</td>
<td>70</td>
<td>74</td>
<td>Northern Africa</td>
<td>Africa</td>
</tr>
<tr>
<td>Libya</td>
<td>6.5</td>
<td>14</td>
<td>2.6</td>
<td>75</td>
<td>72</td>
<td>77</td>
<td>Northern Africa</td>
<td>Africa</td>
</tr>
<tr>
<td>Morocco</td>
<td>32.6</td>
<td>30</td>
<td>2.3</td>
<td>72</td>
<td>70</td>
<td>74</td>
<td>Northern Africa</td>
<td>Africa</td>
</tr>
<tr>
<td>Sth Sudan</td>
<td>9.4</td>
<td>101</td>
<td>5.4</td>
<td>52</td>
<td>50</td>
<td>53</td>
<td>Northern Africa</td>
<td>Africa</td>
</tr>
<tr>
<td>Sudan</td>
<td>33.5</td>
<td>67</td>
<td>4.2</td>
<td>60</td>
<td>58</td>
<td>62</td>
<td>Northern Africa</td>
<td>Africa</td>
</tr>
<tr>
<td>Tunisia</td>
<td>10.8</td>
<td>20</td>
<td>2.1</td>
<td>75</td>
<td>73</td>
<td>77</td>
<td>Northern Africa</td>
<td>Africa</td>
</tr>
<tr>
<td>Benin</td>
<td>9.4</td>
<td>81</td>
<td>5.4</td>
<td>56</td>
<td>54</td>
<td>58</td>
<td>Western Africa</td>
<td>Africa</td>
</tr>
<tr>
<td>Gambia</td>
<td>1.8</td>
<td>70</td>
<td>4.9</td>
<td>58</td>
<td>57</td>
<td>59</td>
<td>Western Africa</td>
<td>Africa</td>
</tr>
<tr>
<td>Ghana</td>
<td>25.5</td>
<td>47</td>
<td>4.2</td>
<td>64</td>
<td>63</td>
<td>65</td>
<td>Western Africa</td>
<td>Africa</td>
</tr>
<tr>
<td>Serbia</td>
<td>7.1</td>
<td>7</td>
<td>1.3</td>
<td>74</td>
<td>71</td>
<td>77</td>
<td>Southern Europe</td>
<td>Europe</td>
</tr>
<tr>
<td>Slovenia</td>
<td>2.1</td>
<td>3</td>
<td>1.5</td>
<td>80</td>
<td>76</td>
<td>83</td>
<td>Southern Europe</td>
<td>Europe</td>
</tr>
<tr>
<td>Spain</td>
<td>46.2</td>
<td>3</td>
<td>1.4</td>
<td>82</td>
<td>79</td>
<td>85</td>
<td>Southern Europe</td>
<td>Europe</td>
</tr>
<tr>
<td>Australia</td>
<td>22.0</td>
<td>4</td>
<td>1.9</td>
<td>82</td>
<td>80</td>
<td>84</td>
<td>Oceania</td>
<td>Oceania</td>
</tr>
<tr>
<td>New Zeal.</td>
<td>4.4</td>
<td>5</td>
<td>2.1</td>
<td>81</td>
<td>79</td>
<td>83</td>
<td>Oceania</td>
<td>Oceania</td>
</tr>
</tbody>
</table>

**Column header contains value:** 2012 ... simple remedy

**Column header contains values:** M, F ... use 2nd table with unit of observation country-year-sex
Using ggplot2 to display gender-specific life-expectancy

```r
p <- ggplot(data=subset(w, area=="Africa"),
             aes(x=reorder(factor(country), leF), y=leF))
p + geom_point(color="red") +
              geom_point(aes(y=leM), color="blue")
```

Data Set Example 3
Data Set Example 3

Using ggplot2 to display gender-specific life-expectancy

```r
p <- ggplot(data=subset(w, area=="Africa"), aes(x=reorder(factor(country), le), y=le, color=sex))
p + geom_point()
```
Data Set Example 4

Tidy?

country  measure      value
Algeria   imr          24
Algeria   tfr          2.9
Algeria   le           73
Egypt     imr          24
Egypt     tfr          2.9
Egypt     le           72
Libya     imr          14
Libya     tfr          2.6
Libya     le           75
Morocco   imr          30
Morocco   tfr          2.3
Morocco   le           72
South Sudan imr        101
South Sudan tfr        5.4
South Sudan le         52

.
Data Set Example 4

Tidy?

<table>
<thead>
<tr>
<th>country</th>
<th>measure</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algeria</td>
<td>imr</td>
<td>24</td>
</tr>
<tr>
<td>Algeria</td>
<td>tfr</td>
<td>2.9</td>
</tr>
<tr>
<td>Algeria</td>
<td>le</td>
<td>73</td>
</tr>
<tr>
<td>Egypt</td>
<td>imr</td>
<td>24</td>
</tr>
<tr>
<td>Egypt</td>
<td>tfr</td>
<td>2.9</td>
</tr>
<tr>
<td>Egypt</td>
<td>le</td>
<td>72</td>
</tr>
<tr>
<td>Libya</td>
<td>imr</td>
<td>14</td>
</tr>
<tr>
<td>Libya</td>
<td>tfr</td>
<td>2.6</td>
</tr>
<tr>
<td>Libya</td>
<td>le</td>
<td>75</td>
</tr>
<tr>
<td>Morocco</td>
<td>imr</td>
<td>30</td>
</tr>
<tr>
<td>Morocco</td>
<td>tfr</td>
<td>2.3</td>
</tr>
<tr>
<td>Morocco</td>
<td>le</td>
<td>72</td>
</tr>
<tr>
<td>South Sudan</td>
<td>imr</td>
<td>101</td>
</tr>
<tr>
<td>South Sudan</td>
<td>tfr</td>
<td>5.4</td>
</tr>
<tr>
<td>South Sudan</td>
<td>le</td>
<td>52</td>
</tr>
</tbody>
</table>

. . .
. . .
. . .
. . .

Each variable is not saved in its own column.
## Data Set Example 5

**Tidy?**

<table>
<thead>
<tr>
<th>country</th>
<th>imr</th>
<th>tfr</th>
<th>le</th>
<th>region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algeria</td>
<td>24</td>
<td>2.9</td>
<td>73</td>
<td>Northern Africa</td>
</tr>
<tr>
<td>Egypt</td>
<td>24</td>
<td>2.9</td>
<td>72</td>
<td>Northern Africa</td>
</tr>
<tr>
<td>Benin</td>
<td>81</td>
<td>5.4</td>
<td>56</td>
<td>Western Africa</td>
</tr>
<tr>
<td>Burkina Faso</td>
<td>65</td>
<td>6.0</td>
<td>55</td>
<td>Western Africa</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>Albania</td>
<td>18</td>
<td>1.4</td>
<td>75</td>
<td>Southern Europe</td>
</tr>
<tr>
<td>Bosnia-Herz.</td>
<td>5</td>
<td>1.2</td>
<td>76</td>
<td>Southern Europe</td>
</tr>
<tr>
<td>Croatia</td>
<td>4</td>
<td>1.5</td>
<td>77</td>
<td>Southern Europe</td>
</tr>
<tr>
<td>Greece</td>
<td>4</td>
<td>1.5</td>
<td>80</td>
<td>Southern Europe</td>
</tr>
<tr>
<td>Italy</td>
<td>3</td>
<td>1.4</td>
<td>82</td>
<td>Southern Europe</td>
</tr>
</tbody>
</table>
Data Set Example 5

Tidy?

<table>
<thead>
<tr>
<th>country</th>
<th>imr</th>
<th>tfr</th>
<th>le</th>
<th>region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algeria</td>
<td>24</td>
<td>2.9</td>
<td>73</td>
<td>Northern Africa</td>
</tr>
<tr>
<td>Egypt</td>
<td>24</td>
<td>2.9</td>
<td>72</td>
<td>Northern Africa</td>
</tr>
<tr>
<td>Benin</td>
<td>81</td>
<td>5.4</td>
<td>56</td>
<td>Western Africa</td>
</tr>
<tr>
<td>Burkina Faso</td>
<td>65</td>
<td>6.0</td>
<td>55</td>
<td>Western Africa</td>
</tr>
<tr>
<td></td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>Albania</td>
<td>18</td>
<td>1.4</td>
<td>75</td>
<td>Southern Europe</td>
</tr>
<tr>
<td>Bosnia-Herz.</td>
<td>5</td>
<td>1.2</td>
<td>76</td>
<td>Southern Europe</td>
</tr>
<tr>
<td>Croatia</td>
<td>4</td>
<td>1.5</td>
<td>77</td>
<td>Southern Europe</td>
</tr>
<tr>
<td>Greece</td>
<td>4</td>
<td>1.5</td>
<td>80</td>
<td>Southern Europe</td>
</tr>
<tr>
<td>Italy</td>
<td>3</td>
<td>1.4</td>
<td>82</td>
<td>Southern Europe</td>
</tr>
</tbody>
</table>

Are multiple variables stored in one column?

Should there be a region column (Northern, Western, Southern, ...) and a separate continent column (Africa, Europe, ...)?
## Data Set Example 6

### Tidy?

<table>
<thead>
<tr>
<th>country</th>
<th>continent</th>
<th>lifeExp</th>
<th>pop</th>
<th>gdpPercap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>41.674</td>
<td>16317921</td>
<td>649.3414</td>
</tr>
<tr>
<td>Albania</td>
<td>Europe</td>
<td>71.581</td>
<td>3326498</td>
<td>2497.4379</td>
</tr>
<tr>
<td>Algeria</td>
<td>Africa</td>
<td>67.744</td>
<td>26298373</td>
<td>5023.2166</td>
</tr>
<tr>
<td>Yemen, Rep.</td>
<td>Asia</td>
<td>55.599</td>
<td>13367997</td>
<td>1879.4967</td>
</tr>
<tr>
<td>Zambia</td>
<td>Africa</td>
<td>46.100</td>
<td>8381163</td>
<td>1210.8846</td>
</tr>
<tr>
<td>Zimbabwe</td>
<td>Africa</td>
<td>60.377</td>
<td>10704340</td>
<td>693.4208</td>
</tr>
</tbody>
</table>

Data Set Example 6 file: world_data_1992.csv

<table>
<thead>
<tr>
<th>country</th>
<th>continent</th>
<th>lifeExp</th>
<th>pop</th>
<th>gdpPercap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>42.129</td>
<td>25268405</td>
<td>726.7341</td>
</tr>
<tr>
<td>Albania</td>
<td>Europe</td>
<td>75.651</td>
<td>3508512</td>
<td>4604.2117</td>
</tr>
<tr>
<td>Algeria</td>
<td>Africa</td>
<td>70.994</td>
<td>31287142</td>
<td>5288.0404</td>
</tr>
<tr>
<td>Yemen, Rep.</td>
<td>Asia</td>
<td>60.308</td>
<td>18701257</td>
<td>2234.8208</td>
</tr>
<tr>
<td>Zambia</td>
<td>Africa</td>
<td>39.193</td>
<td>10595811</td>
<td>1071.6139</td>
</tr>
<tr>
<td>Zimbabwe</td>
<td>Africa</td>
<td>39.989</td>
<td>11926563</td>
<td>672.0386</td>
</tr>
</tbody>
</table>

Data Set Example 6 file: world_data_2002.csv
## Data Set Example 6

### Tidy?

<table>
<thead>
<tr>
<th>country</th>
<th>continent</th>
<th>lifeExp</th>
<th>pop</th>
<th>gdpPercap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>41.674</td>
<td>16317921</td>
<td>649.3414</td>
</tr>
<tr>
<td>Albania</td>
<td>Europe</td>
<td>71.581</td>
<td>3326498</td>
<td>2497.4379</td>
</tr>
<tr>
<td>Algeria</td>
<td>Africa</td>
<td>67.744</td>
<td>26298373</td>
<td>5023.2166</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Yemen, Rep.</td>
<td>Asia</td>
<td>55.599</td>
<td>13367997</td>
<td>1879.4967</td>
</tr>
<tr>
<td>Zambia</td>
<td>Africa</td>
<td>46.100</td>
<td>8381163</td>
<td>1210.8846</td>
</tr>
<tr>
<td>Zimbabwe</td>
<td>Africa</td>
<td>60.377</td>
<td>10704340</td>
<td>693.4208</td>
</tr>
</tbody>
</table>

file: world_data_1992.csv

<table>
<thead>
<tr>
<th>country</th>
<th>continent</th>
<th>lifeExp</th>
<th>pop</th>
<th>gdpPercap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>42.129</td>
<td>25268405</td>
<td>726.7341</td>
</tr>
<tr>
<td>Albania</td>
<td>Europe</td>
<td>75.651</td>
<td>3508512</td>
<td>4604.2117</td>
</tr>
<tr>
<td>Algeria</td>
<td>Africa</td>
<td>70.994</td>
<td>31287142</td>
<td>5288.0404</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Yemen, Rep.</td>
<td>Asia</td>
<td>60.308</td>
<td>18701257</td>
<td>2234.8208</td>
</tr>
<tr>
<td>Zambia</td>
<td>Africa</td>
<td>39.193</td>
<td>10595811</td>
<td>1071.6139</td>
</tr>
<tr>
<td>Zimbabwe</td>
<td>Africa</td>
<td>39.989</td>
<td>11926563</td>
<td>672.0386</td>
</tr>
</tbody>
</table>

file: world_data_2002.csv

The same observational unit (country-year) is stored in multiple files ... and within the data set ... there is a hidden variable value stored in files names.
Tidy Data Summary

Tidy data sets
- Each variable is stored in its own column.
  
  and

- Column headers do not contain values.

  and

- Each observation is stored in its own row.

  and

- Each “type” of observation is stored in a single table.

Messy data sets
- Multiple variables are stored in one column.
- Column headers contain values.
- Variables are stored in both rows and columns
- Multiple types of observational units are stored in the same table
- Single observational unit is stored in multiple tables
Install and Load Packages, Read Data

install.packages("tidyr")
install.packages("dplyr")
install.packages("ggplot2")
install.packages("gapminder")

library("tidyr")
library("dplyr")
library("ggplot2")
library("gapminder")

usle <- read.csv(file="uslifeexp.csv", head=TRUE, sep="\,")

Reshape Data

gather(usle, key="sex", value="lifeexp", le_male, le_female)

```
year  le  le_w  le_wmale  le_wfemale  le_b  le_bmale  le_bfemale  sex  lifeexp
1  1900  47.3  47.6   46.6   48.7   33.0  32.5   33.5 le_male    46.3
2  1901  49.1  49.4   48.0    51.0  33.7    32.2  35.3  le_male    47.6
3  1902  51.5  51.9   50.2   53.8   34.6  32.9   36.4  le_male    49.8
4  1903  50.5  50.9   49.5   52.5   33.1  31.7   34.6  le_male    49.1
   .     .     .        .        .        .       .        .         .
197 1996  76.1  76.8   73.9   79.7   70.2  66.1   74.2  le_female  79.1
198 1997  76.5  77.2   74.3   79.9   71.1  67.2   74.7  le_female  79.4
199 1998  76.7  77.3   74.5   80.0   71.3  67.6   74.8  le_female  79.5
200 1999  76.7  77.3   74.6   79.9   71.4  67.8   74.7  le_female  79.4
```

# "pipe operator" ... think of "then"
select(usle, year, le_male, le_female) %>%
gather(key="sex", value="lifeexp", le_male, le_female) %>% arrange(year)

```
year  sex  lifeexp
1  1900 le_male  46.3
2  1900 le_female  48.3
3  1901 le_male  47.6
4  1901 le_female  50.6
   .     .        .
197 1998 le_male  73.8
198 1998 le_female  79.5
199 1999 le_male  73.9
200 1999 le_female  79.4
```
### Rename and Reshape Data

```r
select(usle, year, le_male, le_female) %>%
  rename(male = le_male, female = le_female) %>%
gather(key="sex", value="lifeexp", male, female) %>%
  arrange(year)
```

<table>
<thead>
<tr>
<th>year</th>
<th>sex</th>
<th>lifeexp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1900</td>
<td>male</td>
<td>46.3</td>
</tr>
<tr>
<td>1900</td>
<td>female</td>
<td>48.3</td>
</tr>
<tr>
<td>1901</td>
<td>male</td>
<td>47.6</td>
</tr>
<tr>
<td>1901</td>
<td>female</td>
<td>50.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1998</td>
<td>male</td>
<td>73.8</td>
</tr>
<tr>
<td>1998</td>
<td>female</td>
<td>79.5</td>
</tr>
<tr>
<td>1999</td>
<td>male</td>
<td>73.9</td>
</tr>
<tr>
<td>1999</td>
<td>female</td>
<td>79.4</td>
</tr>
</tbody>
</table>
```
# EXERCISE:
# Display tidy data that shows
# life expectancy by race (black, white)

# EXTRA CREDIT EXERCISE:
# Using your tidy data along with additional "piping" ...
# display a ggplot2 graph that shows life expectancy by race (black, white)
# EXERCISE:
# Display tidy data that shows
# life expectancy by race (black, white)
select(usle, year, le_b, le_w) %>%
  rename(black = le_b, white = le_w) %>%
gather(key="race", value="lifeexp", black, white) %>%
  arrange(year)

# EXTRA CREDIT EXERCISE:
# Using your tidy data along with additional "piping" ...
# display a ggplot2 graph that shows life expectancy by race (black, white)
select(usle, year, le_b, le_w) %>%
  rename(black = le_b, white = le_w) %>%
gather(key="race", value="lifeexp", black, white) %>%
  arrange(year) %>%
ggplot(aes(year, lifeexp, color= race)) + geom_point()
# show life expectancy for black, white, male, female on same graph
mfle <- select(usle, year, le_male, le_female) %>%
  rename(male = le_male, female = le_female) %>%
gather(key="sex", value="lifeexp", male, female) %>%
arrange(year)

select(usle, year, le_b, le_w) %>%
  rename(black = le_b, white = le_w) %>%
gather(key="race", value="lifeexp", black, white) %>%
arrange(year) %>%
ggplot(aes(year, lifeexp, color= race)) +
  geom_point() +
  geom_point(data = mfle, aes(color = sex )) +
  scale_color_discrete(name="Group")
Add New Columns

# use mutate to store life expectancy gap between male and female, 
# and between black and white

```r
mutate(usle, race_le_gap = le_w - le_b, sex_le_gap = le_female - le_male) %>% 
  select(year, race_le_gap, sex_le_gap)
```

# is above data tidy?

```r
mutate(usle, race_le_gap = le_w - le_b, sex_le_gap = le_female - le_male) %>% 
  select(year, race_le_gap, sex_le_gap) %>% 
  ggplot(aes(year, race_le_gap)) + 
  geom_point(color="red") + 
  geom_point(aes(y=sex_le_gap), color="blue") + 
  ylab("life expectancy gap")
```
# how to tidy this data?

```r
mutate(usle, race = le_w - le_b, sex = le_female - le_male) %>%
  select(year, race, sex) %>%
  gather(key="le_gap_type", value="le_gap_years", race, sex)
```

# and then use tidy data to draw graph?

```r
mutate(usle, race = le_w - le_b, sex = le_female - le_male) %>%
  select(year, race, sex) %>%
  gather(key="le_gap_type", value="le_gap_years", race, sex) %>%
  ggplot(aes(year, le_gap_years, color = le_gap_type)) +
  geom_point() +
  ylab("life expectancy gap")
```
# Display tidy data where unit of observation is year-race-sex

```r
select(usle, year, le_wmale, le_wfemale, le_bmale, le_bfemale) %>%
rename(white_male = le_wmale, white_female = le_wfemale, black_male = le_bmale,
       black_female = le_bfemale) %>%
gather(key="race_sex", value="lifeexp", white_male, white_female, black_male, black_female)
%>% arrange(year, race_sex)
```

```r
select(usle, year, le_wmale, le_wfemale, le_bmale, le_bfemale) %>%
rename(white_male = le_wmale, white_female = le_wfemale, black_male = le_bmale,
       black_female = le_bfemale) %>%
gather(key="race_sex", value="lifeexp", white_male, white_female, black_male, black_female)
%>% arrange(year, race_sex) %>%
ggplot(aes(year, lifeexp, color = race_sex)) +
geom_point()
```
# Reminder - tidy data: each variable is saved in ITS OWN column

```r
select(usle, year, le_wmale, le_wfemale, le_bmale, le_bfemale) %>%
rename(white_male = le_wmale, white_female = le_wfemale, black_male = le_bmale, black_female = le_bfemale) %>%
gather(key="racesex", value="lifeexp", white_male, white_female, black_male, black_female) %>% arrange(year, racesex) %>%
separate(racesex, c("race", "sex"), sep = "_")
```

<table>
<thead>
<tr>
<th>year</th>
<th>race</th>
<th>sex</th>
<th>lifeexp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1900</td>
<td>black</td>
<td>female</td>
</tr>
<tr>
<td>2</td>
<td>1900</td>
<td>black</td>
<td>male</td>
</tr>
<tr>
<td>3</td>
<td>1900</td>
<td>white</td>
<td>female</td>
</tr>
<tr>
<td>4</td>
<td>1900</td>
<td>white</td>
<td>male</td>
</tr>
<tr>
<td>397</td>
<td>1999</td>
<td>black</td>
<td>female</td>
</tr>
<tr>
<td>398</td>
<td>1999</td>
<td>black</td>
<td>male</td>
</tr>
<tr>
<td>399</td>
<td>1999</td>
<td>white</td>
<td>female</td>
</tr>
<tr>
<td>400</td>
<td>1999</td>
<td>white</td>
<td>male</td>
</tr>
</tbody>
</table>

Store Each Variable in its Own Column
Combine Multiple Columns

# tidyr function that goes in reverse direction: unite

tidy_le <- select(usle, year, le_wmale, le_wfemale, le_bmale, le_bfemale) %>%
rename(white_male = le_wmale, white_female = le_wfemale, black_male = le_bmale,
    black_female = le_bfemale) %>%
gather(key="racesex", value="lifeexp", white_male, white_female, black_male, black_female)
%>% arrange(year, racesex) %>%
separate(racesex, c("race", "sex"), sep = "_")

unite(tidy_le, "racesex", c(race, sex), sep = "_")

<table>
<thead>
<tr>
<th>year</th>
<th>racesex</th>
<th>lifeexp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1900 black_female</td>
<td>33.5</td>
</tr>
<tr>
<td>2</td>
<td>1900 black_male</td>
<td>32.5</td>
</tr>
<tr>
<td>3</td>
<td>1900 white_female</td>
<td>48.7</td>
</tr>
<tr>
<td>4</td>
<td>1900 white_male</td>
<td>46.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>397</td>
<td>1999 black_female</td>
<td>74.7</td>
</tr>
<tr>
<td>398</td>
<td>1999 black_male</td>
<td>67.8</td>
</tr>
<tr>
<td>399</td>
<td>1999 white_female</td>
<td>79.9</td>
</tr>
<tr>
<td>400</td>
<td>1999 white_male</td>
<td>74.6</td>
</tr>
</tbody>
</table>
Reverse of `gather()`: `spread()`

# tidyverse function that goes in reverse direction: `spread` (rows -> columns; “long” to “wide”)

```
unite(tidy_le, "racesex", c(race, sex), sep = "_") %>%
spread(key="racesex", value="lifeexp")
```

```
<table>
<thead>
<tr>
<th>year</th>
<th>black_female</th>
<th>black_male</th>
<th>white_female</th>
<th>white_male</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>33.5</td>
<td>32.5</td>
<td>48.7</td>
<td>46.6</td>
</tr>
<tr>
<td>2</td>
<td>35.3</td>
<td>32.2</td>
<td>51.0</td>
<td>48.0</td>
</tr>
<tr>
<td>3</td>
<td>36.4</td>
<td>32.9</td>
<td>53.8</td>
<td>50.2</td>
</tr>
<tr>
<td>4</td>
<td>34.6</td>
<td>31.7</td>
<td>52.5</td>
<td>49.5</td>
</tr>
<tr>
<td>97</td>
<td>74.2</td>
<td>66.1</td>
<td>79.7</td>
<td>73.9</td>
</tr>
<tr>
<td>98</td>
<td>74.7</td>
<td>67.2</td>
<td>79.9</td>
<td>74.3</td>
</tr>
<tr>
<td>99</td>
<td>74.8</td>
<td>67.6</td>
<td>80.0</td>
<td>74.5</td>
</tr>
<tr>
<td>100</td>
<td>74.7</td>
<td>67.8</td>
<td>79.9</td>
<td>74.6</td>
</tr>
</tbody>
</table>
```

unite(tidy_le, "race_sex", c(race, sex), sep = "_") %>%
spread(key="race_sex", value="lifeexp") %>%
rename(le_bfemale = black_female, le_bmale = black_male, le_wfemale = white_female, le_wmale = white_male)
Exercise

# EXERCISE:
# list data in tidy format with group as one variable
# (taking values: all, male, female, black, white, white_male, white_female, black_male, black_female),
# and life_expectancy as the other variable.

<table>
<thead>
<tr>
<th>year</th>
<th>group</th>
<th>life_expectancy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1700</td>
<td>all</td>
<td>47.3</td>
</tr>
<tr>
<td>1701</td>
<td>all</td>
<td>49.1</td>
</tr>
<tr>
<td>1701</td>
<td>black</td>
<td>33.7</td>
</tr>
<tr>
<td>1701</td>
<td>black_female</td>
<td>35.3</td>
</tr>
<tr>
<td>1701</td>
<td>black_male</td>
<td>32.2</td>
</tr>
<tr>
<td>1701</td>
<td>female</td>
<td>50.6</td>
</tr>
<tr>
<td>1701</td>
<td>male</td>
<td>47.6</td>
</tr>
<tr>
<td>1701</td>
<td>white</td>
<td>49.4</td>
</tr>
<tr>
<td>1701</td>
<td>white_female</td>
<td>51.0</td>
</tr>
<tr>
<td>1701</td>
<td>white_male</td>
<td>48.0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
# EXERCISE:
# list data in tidy format with group as one variable
# (taking values:
# all, male, female, black, white, white_male, white_female, black_male, black_female),
# and life_expectancy as the other variable.

tidy_data <- rename(usle, all=le, male=le_male, female=le_female, black=le_b, white=le_w,
                  white_male=le_wmale, white_female=le_wfemale, black_male=le_bmale,
                  black_female=le_bfemale) %>%
gather(key = "group", value = "life_expectancy", 2:10) %>%
arrange(year, group)
# preview a bit more dplyr
rename(usle, all=le, male=le_male, female=le_female, black=le_b, white=le_w,
white_male=le_wmale, white_female=le_wfemale, black_male=le_bmale,
black_female=le_bfemale) %>%
gather(key = "group", value = "life_expectancy", 2:10) %>%
arrange(year, group) %>%
filter(group == "black" | group == "white")

<table>
<thead>
<tr>
<th>year</th>
<th>group</th>
<th>life_expectancy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1900</td>
<td>black</td>
<td>33.0</td>
</tr>
<tr>
<td>1900</td>
<td>white</td>
<td>47.6</td>
</tr>
<tr>
<td>1901</td>
<td>black</td>
<td>33.7</td>
</tr>
<tr>
<td>1901</td>
<td>white</td>
<td>49.4</td>
</tr>
<tr>
<td>1902</td>
<td>black</td>
<td>34.6</td>
</tr>
<tr>
<td>1902</td>
<td>white</td>
<td>51.9</td>
</tr>
<tr>
<td>1903</td>
<td>black</td>
<td>33.1</td>
</tr>
<tr>
<td>1903</td>
<td>white</td>
<td>50.9</td>
</tr>
<tr>
<td>1904</td>
<td>black</td>
<td>30.8</td>
</tr>
<tr>
<td>1904</td>
<td>white</td>
<td>48.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
rename(usle, all=le, male=le_male, female=le_female, black=le_b, white=le_w, 
   white_male=le_wmale, white_female=le_wfemale, 
   black_male=le_bmale, black_female=le_bfemale) %>%
gather(key = "group", value = "life_expectancy", 2:10) %>%
arrange(year, group) %>%
filter(group == "male" | group == "female") %>%
ggplot(aes(year, life_expectancy, color = group)) + geom_point()
Again ... Tidy Data, Filter Rows, then Graph

rename(usle, all=le, male=le_male, female=le_female, black=le_b, white=le_w, 
  white_male=le_wmale, white_female=le_wfemale, 
  black_male=le_bmale, black_female=le_bfemale) %>%
gather(key = "group", value = "life_expectancy", 2:10) %>%
arrange(year, group) %>%
filter(group %in% c("white_male", "white_female", "black_male", "black_female")) %>%
ggplot(aes(year, life_expectancy, color = group)) + geom_point()
Basic dplyr Principles

consistent with tidyr philosophy

- input: dataframe
- output: dataframe

first argument to dplyr commands is a data frame

input data frame is never modified in place ...
may want to save results in a new data frame

commands are optimized for
  - clairty (clean, clear syntax)
  - computation time (written in C++)
dplyr Commands: Verbs

filter() subset observations (rows)

arrange() order observations (rows)

select() subset variables (columns)

rename() change name of variables (column headers)

mutate() add new variables (columns)

group_by() partition observations into groups based on variable values

summarise() collapse each group into a single row of values
Load gapminder tbl_df

# check structure of gapminder data
str(gapminder)

# tbl_df: improved data.frame for which dplyr provides nice methods for high-level inspection
# these methods do something sensible for datasets with many observations and/or variables

gdf <- as.data.frame(gapminder)
str(gdf)

gtdf <- tbl_df(gdf)
str(gtdf)

# high-level inspection of tbl_df

glimpse(gapminder)
Observations: 1,704
Variables: 6

$ country   (fctr) Afghanistan, Afghanistan, Afghanistan, Afghanistan, Afghanistan,...
$ continent (fctr) Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia...
$ pop        (int) 8425333, 9240934, 10267083, 11537966, 13079460, 14880372, 1288181...
$ gdpPercap  (dbl) 779.4453, 820.8530, 853.1007, 836.1971, 739.9811, 786.1134, 978.0...

View(gapminder) # note capital V
<table>
<thead>
<tr>
<th>country</th>
<th>continent</th>
<th>year</th>
<th>lifeExp</th>
<th>pop</th>
<th>gdpPercap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>1952</td>
<td>28.801</td>
<td>8425333</td>
<td>779.4453</td>
</tr>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>1957</td>
<td>30.332</td>
<td>9240934</td>
<td>820.8530</td>
</tr>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>1962</td>
<td>31.997</td>
<td>10267083</td>
<td>853.1007</td>
</tr>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>1967</td>
<td>34.020</td>
<td>11537966</td>
<td>836.1971</td>
</tr>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>1972</td>
<td>36.088</td>
<td>13079460</td>
<td>739.9811</td>
</tr>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>1977</td>
<td>38.438</td>
<td>14880372</td>
<td>786.1134</td>
</tr>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>1982</td>
<td>39.854</td>
<td>12881816</td>
<td>978.0114</td>
</tr>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>1987</td>
<td>40.822</td>
<td>13867957</td>
<td>852.3959</td>
</tr>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>1992</td>
<td>41.674</td>
<td>16317921</td>
<td>649.3414</td>
</tr>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>1997</td>
<td>41.763</td>
<td>22227415</td>
<td>635.3414</td>
</tr>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>2002</td>
<td>42.129</td>
<td>25268405</td>
<td>726.7341</td>
</tr>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>2007</td>
<td>43.828</td>
<td>31889923</td>
<td>974.5803</td>
</tr>
<tr>
<td>Zimbabwe</td>
<td>Africa</td>
<td>1952</td>
<td>48.451</td>
<td>3080907</td>
<td>406.8841</td>
</tr>
<tr>
<td>Zimbabwe</td>
<td>Africa</td>
<td>1957</td>
<td>50.469</td>
<td>3646340</td>
<td>518.7643</td>
</tr>
<tr>
<td>Zimbabwe</td>
<td>Africa</td>
<td>1962</td>
<td>52.358</td>
<td>4277736</td>
<td>527.2722</td>
</tr>
<tr>
<td>Zimbabwe</td>
<td>Africa</td>
<td>1967</td>
<td>53.995</td>
<td>4995432</td>
<td>569.7951</td>
</tr>
<tr>
<td>Zimbabwe</td>
<td>Africa</td>
<td>1972</td>
<td>55.635</td>
<td>5861135</td>
<td>799.3622</td>
</tr>
<tr>
<td>Zimbabwe</td>
<td>Africa</td>
<td>1977</td>
<td>57.674</td>
<td>6642107</td>
<td>685.5877</td>
</tr>
<tr>
<td>Zimbabwe</td>
<td>Africa</td>
<td>1982</td>
<td>60.363</td>
<td>7636524</td>
<td>788.8550</td>
</tr>
<tr>
<td>Zimbabwe</td>
<td>Africa</td>
<td>1987</td>
<td>62.351</td>
<td>9216418</td>
<td>706.1573</td>
</tr>
<tr>
<td>Zimbabwe</td>
<td>Africa</td>
<td>1992</td>
<td>60.377</td>
<td>10704340</td>
<td>693.4208</td>
</tr>
<tr>
<td>Zimbabwe</td>
<td>Africa</td>
<td>1997</td>
<td>46.809</td>
<td>11404948</td>
<td>792.4500</td>
</tr>
<tr>
<td>Zimbabwe</td>
<td>Africa</td>
<td>2002</td>
<td>39.989</td>
<td>11926563</td>
<td>672.0386</td>
</tr>
<tr>
<td>Zimbabwe</td>
<td>Africa</td>
<td>2007</td>
<td>43.487</td>
<td>12311143</td>
<td>469.7093</td>
</tr>
</tbody>
</table>
Subset Observations

filter(gapminder, country == "United States")

filter(gapminder, lifeExp < 30)

filter(gapminder, pop < 1000000 )

filter(gapminder, pop < 1000000, year == 2007)

filter(gapminder, pop < 1000000 & year == 2007)

filter(gapminder, country == "United States" | country == "Canada", year > 2000)

filter(gapminder, country %in% c("United States", "Canada"), year > 2000)

distinct(gapminder, country)
View(distinct(gapminder, country))
distinct(gapminder, country) %>% View()

distinct(as.data.frame(gapminder), country)
Subset Columns

```r
select(gapminder, country, continent)

country_continent <- select(gapminder, country, continent) %>% distinct()
country_continent

select(gapminder, -continent)  # "-" means not ... gives TIDIER data set
tgap <- select(gapminder, -continent)

# But how to combine tgap and country_continent when want
# to summarize values by continent???
# Will later use a "join" function to combine

select(gapminder, year, country, continent, lifeExp)  # select and re-order columns

select(gapminder, starts_with("co"))

select(gapminder, country:lifeExp)  # range
Exercises

# EXERCISE:
# list all countries showing only life expectancy for 2007

# EXTRA CREDIT EXERCISE:
# list all countries showing only life expectancy for 2007
# with life expectancy variable named le (rather than lifeExp)
Exercise Solutions

# EXERCISE:
# list all countries showing only life expectancy for 2007

filter(gapminder, year == 2007) %>% select(country, year, lifeExp)

# EXTRA CREDIT EXERCISE:
# list all countries showing only life expectancy for 2007
# with life expectancy variable named le (rather than lifeExp)

filter(gapminder, year == 2007) %>%
select(country, year, lifeExp) %>%
rename(le = lifeExp)
Order Rows

arrange(gapminder, year)

rename(gapminder, le = lifeExp) %>% filter(year == 2007) %>% select(country, year, le) %>% arrange(le)

rename(gapminder, le = lifeExp) %>% filter(year == 2007) %>% select(country, year, le) %>% arrange(desc(le))  # order by descending le

# to list all rows, can use gapminder as a data frame
rename(as.data.frame(gapminder), le = lifeExp) %>% filter(year == 2007) %>% select(country, year, le) %>% arrange(desc(le))

# list 5 countries with highest life expectancy in 2007
# show country, year, and le
rename(gapminder, le = lifeExp) %>% filter(year == 2007) %>% top_n(5, le) %>% select(country, year, le) %>% arrange(desc(le))  # filter rows again

arrange(desc(le))  # sort top 5
# EXERCISE:
# list 10 lowest life expectancies, with lowest at the top
# HINT: use a second filter command
# EXERCISE:
# list 10 lowest life expectancies, with lowest at the top
# HINT: use a second filter command

rename(gapminder, le=lifeExp) %>%
  filter(year == 2007) %>%
  select(country, year, le) %>%
  arrange(le) %>%  # low to high
  slice(1:10)  # filter rows a second time, by position
Construct New Columns

mutate(gapminder, popMil = round(pop / 1000000, 1), le = round(lifeExp, 1))

Source: local data frame [1,704 x 8]

<table>
<thead>
<tr>
<th>country</th>
<th>continent</th>
<th>year</th>
<th>lifeExp</th>
<th>pop</th>
<th>gdpPercap</th>
<th>popMil</th>
<th>le</th>
</tr>
</thead>
<tbody>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>1952</td>
<td>28.801</td>
<td>8425333</td>
<td>779.4453</td>
<td>8.4</td>
<td>28.8</td>
</tr>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>1957</td>
<td>30.332</td>
<td>9240934</td>
<td>820.8530</td>
<td>9.2</td>
<td>30.3</td>
</tr>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>1962</td>
<td>31.997</td>
<td>10267083</td>
<td>853.1007</td>
<td>10.3</td>
<td>32.0</td>
</tr>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>1967</td>
<td>34.020</td>
<td>11537966</td>
<td>836.1971</td>
<td>11.5</td>
<td>34.0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

transmute(gapminder, country = country, y = year,

   popMil = round(pop / 1000000, 1), le = round(lifeExp, 1)) %>%

arrange(y, country)

Source: local data frame [1,704 x 4]

<table>
<thead>
<tr>
<th>country</th>
<th>y</th>
<th>popMil</th>
<th>le</th>
</tr>
</thead>
<tbody>
<tr>
<td>Afghanistan</td>
<td>1952</td>
<td>8.4</td>
<td>28.8</td>
</tr>
<tr>
<td>Albania</td>
<td>1952</td>
<td>1.3</td>
<td>55.2</td>
</tr>
<tr>
<td>Algeria</td>
<td>1952</td>
<td>9.3</td>
<td>43.1</td>
</tr>
<tr>
<td>Angola</td>
<td>1952</td>
<td>4.2</td>
<td>30.0</td>
</tr>
</tbody>
</table>
Window Functions

# window functions take a vector of n values and return n values
# types of window functions:
#  - ranking and ordering functions
#  - cumulative aggregates
#  - access to previous and next values

filter(gapminder, year == 2007) %>%
  mutate(le_rank = dense_rank(lifeExp)) %>%
  select(country, continent, year, lifeExp, le_rank) %>%
  arrange(le_rank)

# how to assign lowest rank to highest life expectancy?
filter(gapminder, year == 2007) %>%
  mutate(le_rank = dense_rank(-lifeExp)) %>%
  select(country, continent, year, lifeExp, le_rank) %>%
  arrange(le_rank)
### Window Functions: Cumulative Sum

```r
filter(as.data.frame(gapminder), year == 1952) %>%
  arrange(continent, country) %>%
  mutate(popMil = round(pop / 1000000, 1)) %>%
  mutate(cumpopMil = cumsum(popMil))
```

```
country continent year lifeExp  pop    gdpPercap popMil  cumpopMil
1 Algeria    Africa 1952  43.077 9279525 2449.0082  9.3       9.3
2  Angola    Africa 1952  30.015 4232095 3520.6103  4.2      13.5
3   Benin    Africa 1952  38.223 1738315 1062.7522  1.7      15.2
4 Botswana    Africa 1952  47.622  442308  851.2411  0.4      15.6
 .         ...        ...   ...    ...       ...       ...       ...       ...
142 New Zealand Oceania 1952  69.390 1994794 10556.5757 2.0    2406.6
```

```r
filter(as.data.frame(gapminder), year == 2007) %>%
  arrange(continent, country) %>%
  mutate(popMil = round(pop / 1000000, 1)) %>%
  mutate(cumpopMil = cumsum(popMil))
```

```
country continent year lifeExp  pop    gdpPercap popMil  cumpopMil
1 Algeria    Africa 2007  72.301 33333216  6223.3675 33.3     33.3
2  Angola    Africa 2007  42.731 12420476  4797.2313 12.4     45.7
 .         ...        ...   ...    ...       ...       ...       ...       ...
142 New Zealand Oceania 2007  80.204 4115771 25185.0091 4.1   6251.1
```
Group Data: Construct New Column Values By Group

filter(gapminder, year == 2007) %>%
mutate(popMil = round(pop / 1000000, 1)) %>%
arrange(continent, popMil) %>%
group_by(continent) %>%
mutate(cumpopMil = cumsum(popMil)) %>% View()

country continent year lifeExp pop gdpPerCap popMil cumpopMil
(fctr) (fctr) (int) (dbl) (int) (dbl) (dbl) (dbl)
1 Sao Tome and Principe Africa 2007 65.528 199579 1598.4351 0.2 0.2
2 Djibouti Africa 2007 54.791 496374 2082.4816 0.5 0.7
3 Equatorial Guinea Africa 2007 51.579 551201 12154.0897 0.6 1.3
4 Comoros Africa 2007 65.152 710960 986.1479 0.7 2.0
5 Reunion Africa 2007 76.442 798094 7670.1226 0.8 2.8
6 Swaziland Africa 2007 39.613 1133066 4513.4806 1.1 3.9
52 Nigeria Africa 2007 46.869 135031164 2013.9773 135.0 929.6
53 Trinidad and Tobago Americas 2007 69.819 1056608 18008.5092 1.1 1.1

select(gapminder, country, year, pop) %>%
group_by(country) %>%
mutate(pop_lag = lag(pop), pop_chg = pop - pop_lag,
       pop_pctchg = round(pop_chg/pop_lag * 100, 1)) %>% View()
Exercise

# list only rows that experienced a population decline during the previous 5 years
# show country, year, pop, pop_chg, pop_pctchg
Exercise Solution

# list only rows that experienced a population decline during the previous 5 years
# show country, year, pop, pop_chg, pop_pctchg

select(gapminder, country, year, pop) %>%
group_by(country) %>%
mutate(pop_lag = lag(pop), pop_chg = pop - pop_lag,
       pop_pctchg = round(pop_chg/pop_lag * 100, 1)) %>%
filter(pop_chg < 0) %>% View()
Summarise Data

# use summarise() with a summary function to change the unit of observation
# summary functions take a vector of values and return a single value
# very often used with group_by()

filter(gapminder, year == 2007) %>%
  summarise(year = mean(year), n_countries = n(),
             avg_country_le = mean(lifeExp), sd_country_le = sd(lifeExp)) # not used with group_by()

filter(gapminder, year == 2007) %>%
  group_by(continent) %>%
  summarise(avg_country_le = mean(lifeExp))

filter(gapminder, year == 2007) %>%
  group_by(continent) %>%
  summarise(year = mean(year), n_countries = n(),
             avg_country_le = mean(lifeExp), sd_country_le = sd(lifeExp))

filter(gapminder, year == 1952) %>%
  group_by(continent) %>%
  summarise(year = mean(year), n_countries = n(),
             avg_country_le = mean(lifeExp), sd_country_le = sd(lifeExp))
Exercises

# EXERCISE 1:
# show data by continent and years 1952 AND 2007
# list number of countries, avg_country_le, and sd_country_le

# EXERCISE 2:
# By continent and year, show ncountries, avg_country_le, sd_country_le for ALL years of data

# EXTRA CREDIT EXERCISE:
# Make a simple graph that shows avg_country_le over time,
# for each continent
# EXERCISE 1:
# show data by continent and years 1952 AND 2007
# list number of countries, avg_country_le, and sd_country_le
filter(gapminder, year == 1952 | year == 2007) %>%
  group_by(continent, year) %>%
  summarise(ncountries = n(), avg_country_le = mean(lifeExp), sd_country_le = sd(lifeExp))

# EXERCISE 2:
# By continent and year, show ncountries, avg_country_le, sd_country_le for ALL years of data
group_by(gapminder, continent, year) %>%
summarise(ncountries = n(), avg_country_le = mean(lifeExp), sd_country_le = sd(lifeExp)) %>%
View()

# EXTRA CREDIT EXERCISE:
# Make a simple graph that shows avg_country_le over time,
# for each continent
group_by(gapminder, continent, year) %>%
  summarise(avg_country_le = mean(lifeExp)) %>%
ggplot(aes(x = year, y = avg_country_le, color = continent)) +
  geom_line()
summarise() “Peels Off” group_by()

# how many continents each country
# has belonged to over time

```r
group_by(gapminder, country) %>%
  summarise(n_continents = n_distinct(continent))
```

# each summarise() "peels off" one level of group_by()

```r
group_by(gapminder, country) %>%
  summarise(n_continents = n_distinct(continent)) %>%
  summarise(avg_n_continents = mean(n_continents))
```

<table>
<thead>
<tr>
<th>country</th>
<th>n_continents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Afghanistan</td>
<td>1</td>
</tr>
<tr>
<td>Albania</td>
<td>1</td>
</tr>
<tr>
<td>Algeria</td>
<td>1</td>
</tr>
<tr>
<td>Angola</td>
<td>1</td>
</tr>
<tr>
<td>Argentina</td>
<td>1</td>
</tr>
<tr>
<td>Australia</td>
<td>1</td>
</tr>
<tr>
<td>Austria</td>
<td>1</td>
</tr>
<tr>
<td>Bahrain</td>
<td>1</td>
</tr>
<tr>
<td>Bangladesh</td>
<td>1</td>
</tr>
<tr>
<td>Belgium</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>avg_n_continents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
</tbody>
</table>
group_by(gapminder, continent, country) %>%
summarise(avg_le_cc = mean(lifeExp)) %>%
summarise(avg_le_c = mean(avg_le_cc)) %>%
summarise(avg_le = mean(avg_le_c))
More Summary Functions

```r
group_by(gapminder, country) %>%
  summarise(year = first(year), le = first(lifeExp))
```

<table>
<thead>
<tr>
<th>country</th>
<th>year</th>
<th>le</th>
</tr>
</thead>
<tbody>
<tr>
<td>Afghanistan</td>
<td>1952</td>
<td>28.801</td>
</tr>
<tr>
<td>Albania</td>
<td>1952</td>
<td>55.230</td>
</tr>
<tr>
<td>Algeria</td>
<td>1952</td>
<td>43.077</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

```r
group_by(gapminder, country) %>%
  summarise(year = last(year), le = last(lifeExp))
```

<table>
<thead>
<tr>
<th>country</th>
<th>year</th>
<th>le</th>
</tr>
</thead>
<tbody>
<tr>
<td>Afghanistan</td>
<td>1962</td>
<td>31.997</td>
</tr>
<tr>
<td>Albania</td>
<td>1962</td>
<td>64.820</td>
</tr>
<tr>
<td>Algeria</td>
<td>1962</td>
<td>48.303</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

```r
group_by(gapminder, country) %>%
  summarise(year = nth(year, 3), le = nth(lifeExp, 3))
```

<table>
<thead>
<tr>
<th>country</th>
<th>year</th>
<th>le</th>
</tr>
</thead>
<tbody>
<tr>
<td>Afghanistan</td>
<td>1962</td>
<td>31.997</td>
</tr>
<tr>
<td>Albania</td>
<td>1962</td>
<td>64.820</td>
</tr>
<tr>
<td>Algeria</td>
<td>1962</td>
<td>48.303</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
summarise_each()

# Apply one or more functions to one or more columns.
# Grouping variables are always excluded from modification.
# Variables to include or exclude ...
# can be specified in same way as variables are specified for select().
# If variable list is not specified, variable list defaults to all non-grouping variables.

group_by(gapminder, continent, year) %>% summarise_each(funs(min, median, max), lifeExp, pop)

# ... and to re-order columns ... can use select:
group_by(gapminder, continent, year) %>%
summarise_each(funs(min, median, max), lifeExp, pop) %>%
select(continent, year, lifeExp_min, lifeExp_median, lifeExp_max, pop_min, pop_median, pop_max)

<table>
<thead>
<tr>
<th>continent</th>
<th>year</th>
<th>lifeExp_min</th>
<th>lifeExp_median</th>
<th>lifeExp_max</th>
<th>pop_min</th>
<th>pop_median</th>
<th>pop_max</th>
</tr>
</thead>
<tbody>
<tr>
<td>(fctr)</td>
<td>(int)</td>
<td>(dbl)</td>
<td>(dbl)</td>
<td>(dbl)</td>
<td>(int)</td>
<td>(dbl)</td>
<td>(int)</td>
</tr>
<tr>
<td>1 Africa</td>
<td>1952</td>
<td>30.000</td>
<td>38.8330</td>
<td>52.724</td>
<td>60011</td>
<td>2668125</td>
<td>33119096</td>
</tr>
<tr>
<td>2 Africa</td>
<td>1957</td>
<td>31.570</td>
<td>40.5925</td>
<td>58.089</td>
<td>61325</td>
<td>2885791</td>
<td>37173340</td>
</tr>
<tr>
<td>3 Africa</td>
<td>1962</td>
<td>32.767</td>
<td>42.6305</td>
<td>60.246</td>
<td>65345</td>
<td>3145210</td>
<td>41871351</td>
</tr>
<tr>
<td>4 Africa</td>
<td>1967</td>
<td>34.113</td>
<td>44.6985</td>
<td>61.557</td>
<td>70787</td>
<td>3473693</td>
<td>47287752</td>
</tr>
</tbody>
</table>
Graphing Results of summarise_each() Functions

# graph min, median, max country life expectancy, by continent

group_by(gapminder, continent, year) %>%
summarise_each(funs(min, median, max), lifeExp, pop) %>%
ggplot(aes(x=year, y = lifeExp_median)) + geom_line() +
geom_line(aes(y = lifeExp_min), linetype = "dashed") +
geom_line(aes(y = lifeExp_max), linetype = "dashed") +
facet_grid(continent ~ .) +
labs(y="Life Expectancy for Countries: min, median, max",
x="")
count() function

# count() function wraps up the
# common combination of group_by() and summarise()

# How many rows for each value of continent?
count(gapminder, continent)

# How many rows for each value of continent and year
count(gapminder, continent, year) %>% View()

filter(gapminder, year == 2002 | year == 2007) %>%
count(continent) # How many rows for each continent, for only years 2002 and 2007

filter(gapminder, year == 2002 | year == 2007) %>%
count(continent, wt = year-2000) # wt is a multiplier
# Here, values for 2002 are multiplied by 2;
# and values for 2007 are multiplied by 5
# (example: 4 Oceania rows ... (2 * 2) + (2 * 7)
dplyr Commands to Combine/Compare Data Sets

- `left_join()`
- `right_join()` potentially add variables (columns) to data sets, making them wider
- `inner_join()`
- `full_join()`
- `bind_rows()` adds observations (rows) to data sets, making them longer
- `semi_join()` potentially remove observations (rows) from data sets, making them shorter
- `anti_join()`
Combining Data Sets: left_join()

# saw above that every country has been associated with just one continent during time period
# so ...
# continent belongs in a table where unit of observation is country
# other variables belong in a table where unit of observation is country-year:

country_continent <- select(gapminder, country, continent) %>% distinct()
country_continent
tgap <- select(gapminder, -continent)
tgap

# BUT descriptive exploration has required
# continent be included in data set for grouping
# HOW TO COMBINE ("join" or "merge") tgap and country_continent?

# join matching rows from second data set to first
left_join(tgap, country_continent, by = "country") %>% View()}
Combining Data Sets: `left_join()` and `right_join()`

```r
country_continent_inc <- slice(country_continent, 6:142)  # cut out rows 1-5
View(country_continent_inc)

tgap_inc <- slice(tgap, 49:144)  # cut out rows 1-48 and rows 145-1704
View(tgap_inc)

# join matching rows from 2nd data set to first
left_join(tgap, country_continent_inc, by = "country") %>% View()

# join matching rows from first data set to 2nd
right_join(tgap, country_continent, by = "country") %>% View()

right_join(tgap, country_continent_inc, by = "country") %>% View()
# country_continent_inc is the driver!
```
Combining Data Sets

`inner_join()`, `full_join()`, `semi_join()`, `anti_join()`

# join and retain only rows in both data sets
inner_join(tgap_inc, country_continent_inc, by = "country") %>% View()

# join and retain all values, all rows
full_join(tgap_inc, country_continent_inc, by = "country") %>% View()

# retain all rows in first data set that have a match in second data set
# (but don't add columns)
semi_join(tgap_inc, country_continent_inc, by = "country") %>% View()

# retain all rows in first data set that do not have a match in second data set
# (but don't add columns)
anti_join(tgap_inc, country_continent_inc, by = "country") %>% View()
Appending Data Sets

# TO APPEND ROWS use bind_rows() ... more efficient than rbind()

tgap1992 <- filter(tgap, year == 1992) %>% select(-year)
tgap1997 <- filter(tgap, year == 1997) %>% select(-year)
tgap2002 <- filter(tgap, year == 2002) %>% select(-year)
tgap2007 <- filter(tgap, year == 2007) %>% select(-year)

tgap1992
tgap2007

bind_rows(tgap1992, tgap1997, tgap2002, tgap2007) %>% View() # ... OOPS .. not quite right!

bind_rows(list(tgap1992, tgap1997, tgap2002, tgap2007), .id="id") %>% View() # ... a bit better!