Data Wrangling

Jameson Watts, Ph.D.

Agenda

1. Review (and upgrade)
2. Joining together
3. Mutation
4. Strings
5. Bringing it all together

Review

Group_by()

The group_by() function doesn't actually do anything visible to a dataframe.

Grouping by province shows the same data as…

```
wine %>%
  group_by(province)
```

```
## # A tibble: 120,975 x 14
## # Groups:   province [423]
##       X1 country description designation points price province region_1
##    <dbl> <chr>   <chr>       <chr>        <dbl> <dbl> <chr>    <chr>
##  1     1 Portug… This is ri… Avidagos        87    15 Douro    <NA>
##  2     2 US      Tart and s… <NA>            87    14 Oregon   Willame…
##  3     3 US      Pineapple … Reserve La…     87    13 Michigan Lake Mi…
##  4     4 US      Much like … Vintner's …     87    65 Oregon   Willame…
##  5     5 Spain   Blackberry… Ars In Vit…     87    15 Norther… Navarra
##  6     6 Italy   Here's a b… Belsito         87    16 Sicily … Vittoria
##  7     7 France  This dry a… <NA>            87    24 Alsace   Alsace
##  8     8 Germany Savory dri… Shine           87    12 Rheinhe… <NA>
##  9     9 France  This has g… Les Natures     87    27 Alsace   Alsace
## 10    10 US      Soft, supp… Mountain C…     87    19 Califor… Napa Va…
## # … with 120,965 more rows, and 6 more variables: region_2 <chr>,
## #   taster_name <chr>, taster_twitter_handle <chr>, title <chr>,
## #   variety <chr>, winery <chr>
```

Group_by()

…grouping by both province and variety.

```
wine %>%
  group_by(province, variety)
```

```
## # A tibble: 120,975 x 14
## # Groups:   province, variety [4,044]
##       X1 country description designation points price province region_1
##    <dbl> <chr>   <chr>       <chr>        <dbl> <dbl> <chr>    <chr>
##  1     1 Portug… This is ri… Avidagos        87    15 Douro    <NA>
##  2     2 US      Tart and s… <NA>            87    14 Oregon   Willame…
##  3     3 US      Pineapple … Reserve La…     87    13 Michigan Lake Mi…
##  4     4 US      Much like … Vintner's …     87    65 Oregon   Willame…
##  5     5 Spain   Blackberry… Ars In Vit…     87    15 Norther… Navarra
##  6     6 Italy   Here's a b… Belsito         87    16 Sicily … Vittoria
##  7     7 France  This dry a… <NA>            87    24 Alsace   Alsace
##  8     8 Germany Savory dri… Shine           87    12 Rheinhe… <NA>
##  9     9 France  This has g… Les Natures     87    27 Alsace   Alsace
## 10    10 US      Soft, supp… Mountain C…     87    19 Califor… Napa Va…
## # … with 120,965 more rows, and 6 more variables: region_2 <chr>,
## #   taster_name <chr>, taster_twitter_handle <chr>, title <chr>,
## #   variety <chr>, winery <chr>
```

Group_by() with summarize()

However, the grouping function sets things up in the background for your summary operations.

```
wine %>%
  summarize('avg. price'=mean(price))
```

```
## # A tibble: 1 x 1
##   `avg. price`
##          <dbl>
## 1         35.4
```

```
wine %>%
  group_by(country) %>%
  summarize('avg. price'=mean(price)) %>%
  head(5)
```

```
## # A tibble: 5 x 2
##   country                `avg. price`
##   <chr>                         <dbl>
## 1 Argentina                      24.5
## 2 Armenia                        14.5
## 3 Australia                      35.4
## 4 Austria                        30.8
## 5 Bosnia and Herzegovina         12.5
```

Exercise

What are the top five varieties in Argentinia by points?

Hint: Use filter(), group_by(), summarize() and top_n() to find your answer

Solution

```
wine %>%
  filter(country=="Argentina") %>%
  group_by(variety) %>%
  summarize(
    'avg_points' = mean(points)
    ) %>%
  top_n(5,avg_points) %>%
  arrange(desc(avg_points))
```

```
## # A tibble: 5 x 2
##   variety               avg_points
##   <chr>                      <dbl>
## 1 Malbec-Cabernet             91.7
## 2 Cabernet Franc-Malbec       91
## 3 Malbec-Petit Verdot         91
## 4 Syrah-Viognier              90.7
## 5 Malbec Blend                90.5
```

Advanced summarize()

Each call to summarize removes one level of grouping.

Note that by running summarize twice, I am back to the results from the previous slide.

```
wine %>%
  filter(country=="Argentina") %>%
  group_by(variety, winery) %>%
  summarize(points = mean(points)) %>%
  summarize('avg_points' = mean(points)) %>%
  top_n(5,avg_points) %>%
  arrange(desc(avg_points))
```

```
## # A tibble: 5 x 2
##   variety               avg_points
##   <chr>                      <dbl>
## 1 Malbec-Cabernet             91.7
## 2 Cabernet Franc-Malbec       91
## 3 Malbec-Petit Verdot         91
## 4 Syrah-Viognier              90.7
## 5 Malbec Blend                90.5
```

Exercise

1. If needed, open the help file on dplyr::summarize()
2. use summarize() with n_distinct()
3. to find the number of unique wineries in Argentina

Solution

```
wine %>%
  filter(country=="Argentina") %>%
  summarize(winery_count=n_distinct(winery))
```

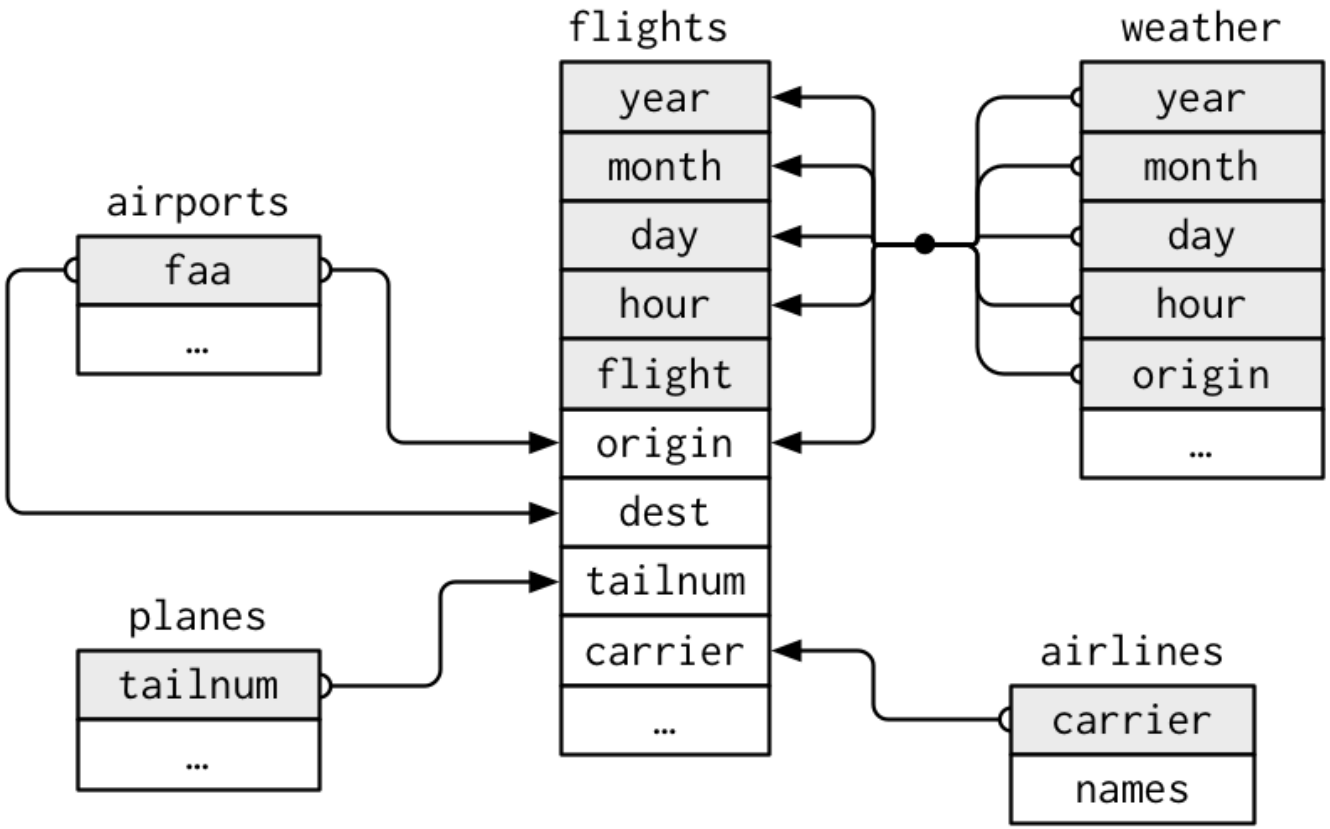
```
## # A tibble: 1 x 1
##   winery_count
##          <int>
## 1          528
```

```
wine %>%
  filter(country=="Argentina") %>%
  count(winery)
```

```
## # A tibble: 528 x 2
##    winery                n
##    <chr>             <int>
##  1 2 Copas               1
##  2 25 Lagunas            1
##  3 Achaval-Ferrer       18
##  4 Aconcagua             3
##  5 Aconga                5
##  6 Acordeón              6
##  7 Adoquin               2
##  8 Aguijón De Abeja      4
##  9 Aitor Ider Balbo      1
## 10 Alamos               24
## # … with 518 more rows
```

Joining Together

Relational data



Definitions

Primary keys: uniquely identifies row in its own dataframe

Foreign keys: uniquely identifies row in another dataframe

Joining up with World Bank data

```
pop <- read_csv("../resources/population.csv")
pop
```

```
## # A tibble: 263 x 61
##    `Country Name` `Country Code` `1960` `1961` `1962` `1963` `1964` `1965`
##    <chr>          <chr>           <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
##  1 Aruba          ABW            5.42e4 5.54e4 5.62e4 5.67e4 5.70e4 5.74e4
##  2 Afghanistan    AFG            9.00e6 9.17e6 9.35e6 9.54e6 9.74e6 9.96e6
##  3 Angola         AGO            5.45e6 5.53e6 5.61e6 5.68e6 5.74e6 5.77e6
##  4 Albania        ALB            1.61e6 1.66e6 1.71e6 1.76e6 1.81e6 1.86e6
##  5 Andorra        AND            1.34e4 1.44e4 1.54e4 1.64e4 1.75e4 1.85e4
##  6 Arab World     ARB            9.22e7 9.47e7 9.73e7 1.00e8 1.03e8 1.06e8
##  7 United Arab E… ARE            9.24e4 1.01e5 1.12e5 1.25e5 1.38e5 1.50e5
##  8 Argentina      ARG            2.05e7 2.08e7 2.12e7 2.15e7 2.18e7 2.22e7
##  9 Armenia        ARM            1.87e6 1.94e6 2.01e6 2.08e6 2.15e6 2.21e6
## 10 American Samoa ASM            2.01e4 2.06e4 2.13e4 2.20e4 2.29e4 2.37e4
## # … with 253 more rows, and 53 more variables: `1966` <dbl>, `1967` <dbl>,
## #   `1968` <dbl>, `1969` <dbl>, `1970` <dbl>, `1971` <dbl>, `1972` <dbl>,
## #   `1973` <dbl>, `1974` <dbl>, `1975` <dbl>, `1976` <dbl>, `1977` <dbl>,
## #   `1978` <dbl>, `1979` <dbl>, `1980` <dbl>, `1981` <dbl>, `1982` <dbl>,
## #   `1983` <dbl>, `1984` <dbl>, `1985` <dbl>, `1986` <dbl>, `1987` <dbl>,
## #   `1988` <dbl>, `1989` <dbl>, `1990` <dbl>, `1991` <dbl>, `1992` <dbl>,
## #   `1993` <dbl>, `1994` <dbl>, `1995` <dbl>, `1996` <dbl>, `1997` <dbl>,
## #   `1998` <dbl>, `1999` <dbl>, `2000` <dbl>, `2001` <dbl>, `2002` <dbl>,
## #   `2003` <dbl>, `2004` <dbl>, `2005` <dbl>, `2006` <dbl>, `2007` <dbl>,
## #   `2008` <dbl>, `2009` <dbl>, `2010` <dbl>, `2011` <dbl>, `2012` <dbl>,
## #   `2013` <dbl>, `2014` <dbl>, `2015` <dbl>, `2016` <dbl>, `2017` <dbl>,
## #   `2018` <dbl>
```

Let's gather, then start with the most recent year

```
pop <- read_csv("../resources/population.csv") %>%
  gather(key="year", value = "population", 3:61) %>%
  rename("country"="Country Name")

pop2017 <- pop %>%
  filter(year==2017) %>%
  select(country, population)

pop2017
```

```
## # A tibble: 263 x 2
##    country              population
##    <chr>                     <dbl>
##  1 Aruba                    105366
##  2 Afghanistan            36296400
##  3 Angola                 29816748
##  4 Albania                 2873457
##  5 Andorra                   77001
##  6 Arab World            411898965
##  7 United Arab Emirates    9487203
##  8 Argentina              44044811
##  9 Armenia                 2944809
## 10 American Samoa            55620
## # … with 253 more rows
```

Try joining with wine on country

```
wine_pop <- wine %>%
  inner_join(pop2017) %>%
  select(country, population, title)

wine_pop
```

```
## # A tibble: 66,569 x 3
##    country   population title
##    <chr>          <dbl> <chr>
##  1 Portugal    10300300 Quinta dos Avidagos 2011 Avidagos Red (Douro)
##  2 Spain       46593236 Tandem 2011 Ars In Vitro Tempranillo-Merlot (Navar…
##  3 Italy       60536709 Terre di Giurfo 2013 Belsito Frappato (Vittoria)
##  4 France      66865144 Trimbach 2012 Gewurztraminer (Alsace)
##  5 Germany     82657002 Heinz Eifel 2013 Shine Gewürztraminer (Rheinhessen)
##  6 France      66865144 Jean-Baptiste Adam 2012 Les Natures Pinot Gris (Al…
##  7 France      66865144 Leon Beyer 2012 Gewurztraminer (Alsace)
##  8 Germany     82657002 Richard Böcking 2013 Devon Riesling (Mosel)
##  9 Argentina   44044811 Felix Lavaque 2010 Felix Malbec (Cafayate)
## 10 Argentina   44044811 Gaucho Andino 2011 Winemaker Selection Malbec (Men…
## # … with 66,559 more rows
```

But wait… we started with over 120k observations and now we're down to 66k?

Did we get everything?

Let's try this again.

```
wine_pop <- wine %>%
  left_join(pop2017) %>%
  select(country, population, title)

wine_pop
```

```
## # A tibble: 120,975 x 3
##    country  population title
##    <chr>         <dbl> <chr>
##  1 Portugal   10300300 Quinta dos Avidagos 2011 Avidagos Red (Douro)
##  2 US               NA Rainstorm 2013 Pinot Gris (Willamette Valley)
##  3 US               NA St. Julian 2013 Reserve Late Harvest Riesling (Lake…
##  4 US               NA Sweet Cheeks 2012 Vintner's Reserve Wild Child Bloc…
##  5 Spain      46593236 Tandem 2011 Ars In Vitro Tempranillo-Merlot (Navarr…
##  6 Italy      60536709 Terre di Giurfo 2013 Belsito Frappato (Vittoria)
##  7 France     66865144 Trimbach 2012 Gewurztraminer (Alsace)
##  8 Germany    82657002 Heinz Eifel 2013 Shine Gewürztraminer (Rheinhessen)
##  9 France     66865144 Jean-Baptiste Adam 2012 Les Natures Pinot Gris (Als…
## 10 US               NA Kirkland Signature 2011 Mountain Cuvée Cabernet Sau…
## # … with 120,965 more rows
```

…hmmm looks like the US isn't matching. But we'll come back to that.

Types of joins

- Mutating
  - inner_join()
  - left_join()
  - right_join()
  - full_join()
- Filtering
  - semi_join()
  - anti_join()

What would be the result if I performed an anti_join() with my wine and population dataframes?

Anti_join

```
wine %>%
  anti_join(pop2017)
```

```
## # A tibble: 54,406 x 14
##       X1 country description designation points price province region_1
##    <dbl> <chr>   <chr>       <chr>        <dbl> <dbl> <chr>    <chr>
##  1     2 US      Tart and s… <NA>            87    14 Oregon   Willame…
##  2     3 US      Pineapple … Reserve La…     87    13 Michigan Lake Mi…
##  3     4 US      Much like … Vintner's …     87    65 Oregon   Willame…
##  4    10 US      Soft, supp… Mountain C…     87    19 Califor… Napa Va…
##  5    12 US      Slightly r… <NA>            87    34 Califor… Alexand…
##  6    14 US      Building o… <NA>            87    12 Califor… Central…
##  7    19 US      Red fruit … <NA>            87    32 Virginia Virginia
##  8    20 US      Ripe aroma… Vin de Mai…     87    23 Virginia Virginia
##  9    21 US      A sleek mi… <NA>            87    20 Oregon   Oregon
## 10    23 US      This wine … Signature …     87    22 Califor… Paso Ro…
## # … with 54,396 more rows, and 6 more variables: region_2 <chr>,
## #   taster_name <chr>, taster_twitter_handle <chr>, title <chr>,
## #   variety <chr>, winery <chr>
```

Mutation

Mutation for new variables

The function mutate() from dplyr allows you to create new variables from existing ones.

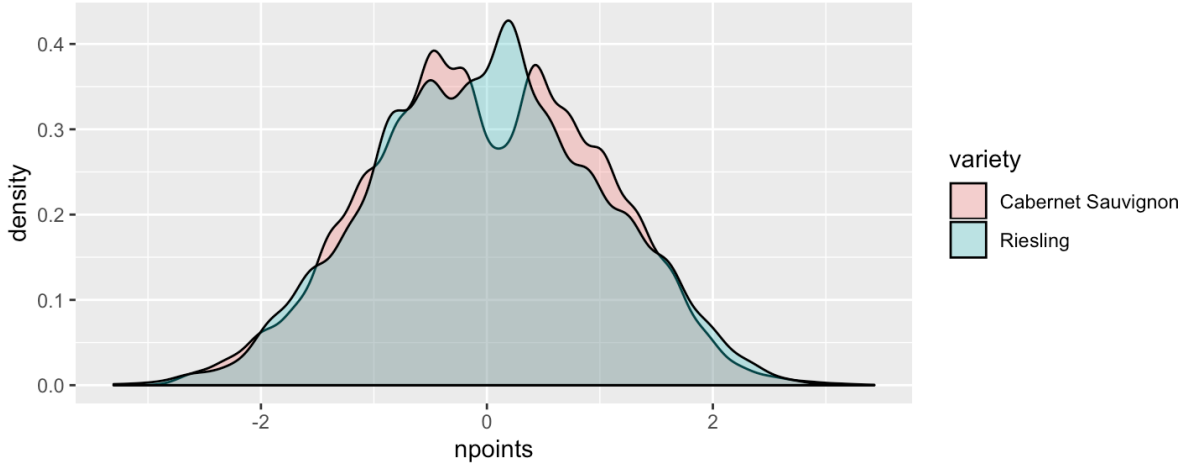
```
wine %>%
  mutate(lprice=log(price)) %>%
  select(price, lprice)
```

```
## # A tibble: 120,975 x 2
##    price lprice
##    <dbl>  <dbl>
##  1    15   2.71
##  2    14   2.64
##  3    13   2.56
##  4    65   4.17
##  5    15   2.71
##  6    16   2.77
##  7    24   3.18
##  8    12   2.48
##  9    27   3.30
## 10    19   2.94
## # … with 120,965 more rows
```

Grouped mutation

You can also mutate by group. For instance, let's say you wanted to standardize prices by variety

```
wine %>%
  group_by(variety) %>%
  mutate(npoints = (points-mean(points)) / sd(points)) %>%
  filter(variety == "Riesling" | variety == "Cabernet Sauvignon") %>%
  ggplot(aes(npoints, fill=variety))+
    geom_density(alpha=0.3)
```



Conditional mutation

Let's change that pesky 'US' identifier to 'United States'

```
wine <- wine %>%
  mutate(country=ifelse(country=="US","United States",country))
wine
```

```
## # A tibble: 120,975 x 14
##       X1 country description designation points price province region_1
##    <dbl> <chr>   <chr>       <chr>        <dbl> <dbl> <chr>    <chr>
##  1     1 Portug… This is ri… Avidagos        87    15 Douro    <NA>
##  2     2 United… Tart and s… <NA>            87    14 Oregon   Willame…
##  3     3 United… Pineapple … Reserve La…     87    13 Michigan Lake Mi…
##  4     4 United… Much like … Vintner's …     87    65 Oregon   Willame…
##  5     5 Spain   Blackberry… Ars In Vit…     87    15 Norther… Navarra
##  6     6 Italy   Here's a b… Belsito         87    16 Sicily … Vittoria
##  7     7 France  This dry a… <NA>            87    24 Alsace   Alsace
##  8     8 Germany Savory dri… Shine           87    12 Rheinhe… <NA>
##  9     9 France  This has g… Les Natures     87    27 Alsace   Alsace
## 10    10 United… Soft, supp… Mountain C…     87    19 Califor… Napa Va…
## # … with 120,965 more rows, and 6 more variables: region_2 <chr>,
## #   taster_name <chr>, taster_twitter_handle <chr>, title <chr>,
## #   variety <chr>, winery <chr>
```

Strings

String basics

```
library(stringr)

name <- "Jameson Watts"
name
```

```
## [1] "Jameson Watts"
```

```
quote <- '"As soon as you stop wanting something, you get it." - Andy Warhol'
quote
```

```
## [1] "\"As soon as you stop wanting something, you get it.\" - Andy Warhol"
```

```
writeLines(quote)
```

```
## "As soon as you stop wanting something, you get it." - Andy Warhol
```

String Operations

```
str_length(quote)
```

```
## [1] 66
```

```
name_quote <- str_c("Name: ", name, "\nQuote: ", quote)
writeLines(name_quote)
```

```
## Name: Jameson Watts
## Quote: "As soon as you stop wanting something, you get it." - Andy Warhol
```

Other common string functions

```
str_sub(name_quote,10,30)
```

```
## [1] "eson Watts\nQuote: \"As"
```

```
str_to_lower(name_quote)
```

```
## [1] "name: jameson watts\nquote: \"as soon as you stop wanting something, you get it.\" - andy warhol"
```

```
str_locate(name_quote,"Jameson")
```

```
##      start end
## [1,]     7  13
```

Regular expressions

```
library(htmlwidgets)
str_detect(name_quote, "on Wa")
```

```
## [1] TRUE
```

```
str_detect(name_quote, "on wa")
```

```
## [1] FALSE
```

```
str_view_all(name_quote, "ou")
```

- Name: Jameson Watts Quote: "As soon as you stop wanting something, you get it." - Andy Warhol

More complicated Regex

Placeholders and Repetition

- . maches any character
- * matches 0 or more
- + matches 1 or more
- ? matches 0 or 1

Anchors

- ^ matches start of string
- $ matches end of string

Character classes

- \d matches any digit.
- \s matches any whitespace (e.g. space, tab, newline).
- [abc] matches a, b, or c.
- [^abc] matches anything except a, b, or c.

Some examples

```
str_view_all(name_quote, "a.+s")
```

- `Name: Jameson Watts` Quote: "`As soon as you stop wanting s`omething, you get it." – Andy Warhol

```
wine %>%
  filter(str_detect(variety,"[Cc]abernet")) %>%
  select(points, price, variety) %>%
  arrange(desc(price))
```

```
## # A tibble: 11,582 x 3
##    points price variety
##     <dbl> <dbl> <chr>
##  1     94   625 Cabernet Sauvignon
##  2     98   625 Cabernet Sauvignon
##  3     97   625 Cabernet Sauvignon
##  4     92   600 Tempranillo-Cabernet Sauvignon
##  5     93   500 Cabernet Sauvignon
##  6     89   500 Tempranillo-Cabernet Sauvignon
##  7     95   500 Cabernet Sauvignon
##  8     89   500 Tempranillo-Cabernet Sauvignon
##  9     92   400 Cabernet Sauvignon
## 10     96   400 Cabernet Sauvignon
## # … with 11,572 more rows
```

Exercise

1. Use filter() and str_detect() to
2. find all Oregon wines
3. with the words ‘espresso’ and ‘black currant’ in their description
4. showing variety, price, points and winery

Note: these sorts of queries can tax your computer

Solution

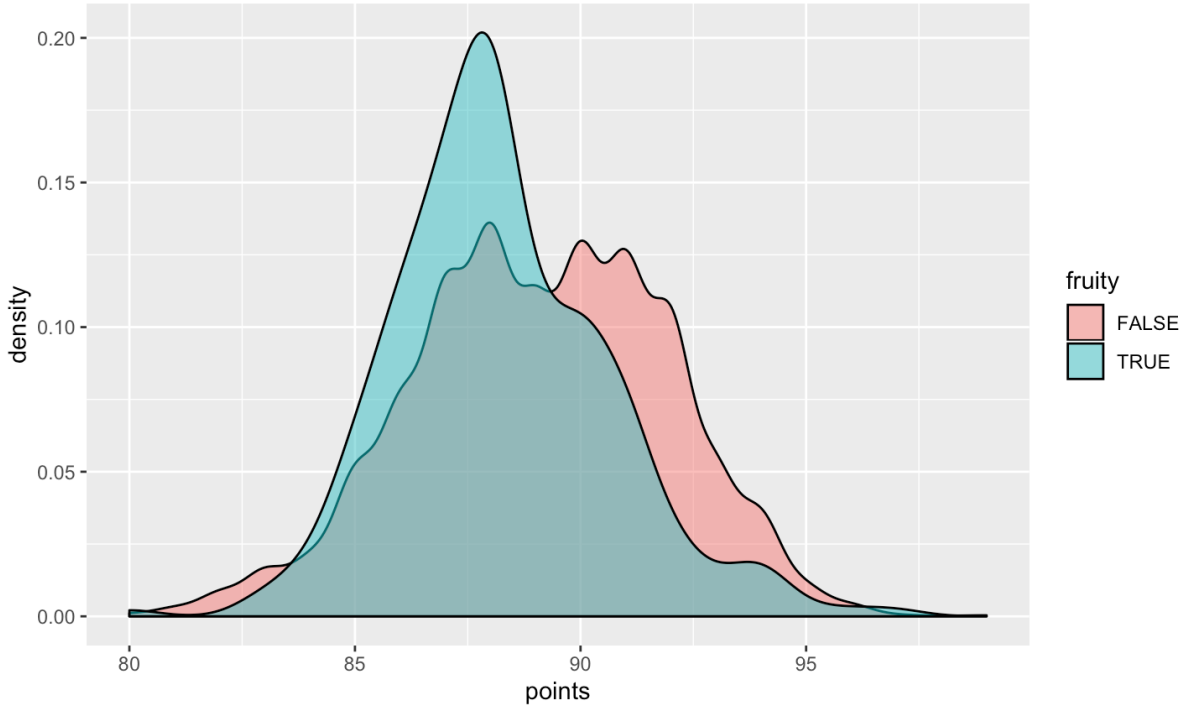
```
wine %>%
  filter(str_detect(description,"[Ee]spresso") & str_detect(description,"[Bb]lack currant")) %>%
  select(variety, price, points, winery) %>%
  arrange(desc(points), price)
```

```
## # A tibble: 80 x 4
##    variety                 price points winery
##    <chr>                   <dbl>  <dbl> <chr>
##  1 Red Blend                  66     95 Podere Sapaio
##  2 Cabernet Sauvignon        100     95 DAOU
##  3 Pinot Noir                 66     94 Shea
##  4 Red Blend                 175     94 Tenuta di Biserno
##  5 Merlot                    400     94 Tua Rita
##  6 Cabernet Sauvignon         29     93 Oso Libre
##  7 Rhône-style Red Blend      38     93 Kaleidos
##  8 Syrah                      43     93 Carlisle
##  9 Cabernet Sauvignon         48     93 Boekenoogen
## 10 Bordeaux-style Red Blend   50     93 Jada Vineyard & Winery
## # … with 70 more rows
```

Bringing it all together

Combining with mutate

```
wine %>%
  filter(province=="Oregon") %>%
  mutate(fruity = str_detect(description,"[Ff]ruity")) %>%
  ggplot(aes(points, fill=fruity))+
    geom_density(alpha=.5)
```



Extracting year from title

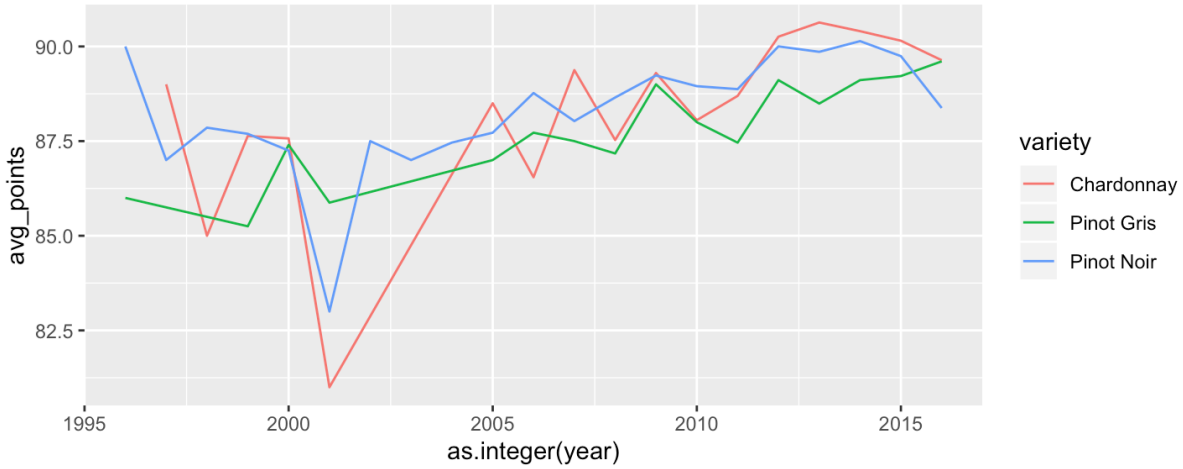
```
wine <- wine %>%
  mutate(year = str_extract(title,"(\\d{4})"))

wine %>%
  select(title, year)
```

```
## # A tibble: 120,975 x 2
##    title                                                          year
##    <chr>                                                          <chr>
##  1 Quinta dos Avidagos 2011 Avidagos Red (Douro)                  2011
##  2 Rainstorm 2013 Pinot Gris (Willamette Valley)                  2013
##  3 St. Julian 2013 Reserve Late Harvest Riesling (Lake Michigan Shor… 2013
##  4 Sweet Cheeks 2012 Vintner's Reserve Wild Child Block Pinot Noir (… 2012
##  5 Tandem 2011 Ars In Vitro Tempranillo-Merlot (Navarra)          2011
##  6 Terre di Giurfo 2013 Belsito Frappato (Vittoria)               2013
##  7 Trimbach 2012 Gewurztraminer (Alsace)                          2012
##  8 Heinz Eifel 2013 Shine Gewürztraminer (Rheinhessen)            2013
##  9 Jean-Baptiste Adam 2012 Les Natures Pinot Gris (Alsace)        2012
## 10 Kirkland Signature 2011 Mountain Cuvée Cabernet Sauvignon (Napa V… 2011
## # … with 120,965 more rows
```

Graphing points by year

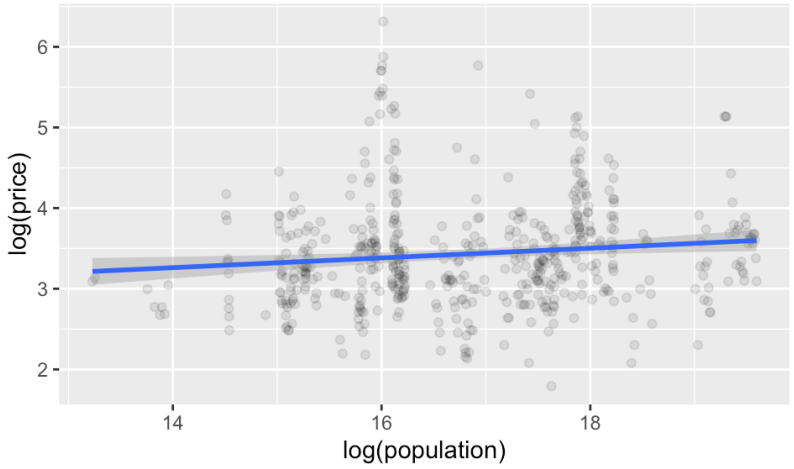
```
wine %>%
  filter(province=="Oregon") %>%
  filter(variety=="Pinot Noir" | variety=="Chardonnay" | variety=="Pinot Gris") %>%
  filter(year >= 1995) %>%
  group_by(year, variety) %>%
  summarize(avg_points = mean(points)) %>%
  ggplot(aes(x=as.integer(year), y=avg_points, color=variety)) +
    geom_line()
```



Joining on more than one column

Now that we've extracted year from the title, we can do a join on both country and year

```
wine %>%
  left_join(pop) %>%
  filter(!is.na(population) & !is.na(year)) %>%
  filter(population < 1000000000) %>%
  group_by(country,year) %>%
  summarize(population=mean(population),price = mean(price)) %>%
  ggplot(aes(x=log(population), y=log(price))) +
    geom_jitter(alpha=.1)+
    geom_smooth(method = lm)
```



Long Exercise

1. Go to https://data.worldbank.org/indicator
2. Find some cool country data
3. Merge it with the wine data
4. Decide on research question
5. Draw a cool graph that answers the question

Bonus: joining with yourself

```
top_wineries <- wine %>%
  group_by(winery) %>%
  summarize(
    avg_points=mean(points),
    count=n()) %>%
  filter(count > 10) %>%
  top_frac(.05,avg_points) %>%
  left_join(wine)

top_wineries
```

```
## # A tibble: 4,065 x 17
##    winery avg_points count    X1 country description designation points
##    <chr>       <dbl> <int> <dbl> <chr>   <chr>       <chr>        <dbl>
##  1 Abeja        92.4    28  6061 United… All variet… Heather Hi…     89
##  2 Abeja        92.4    28  6737 United… Fresh and … <NA>            94
##  3 Abeja        92.4    28 28201 United… The wine i… <NA>            91
##  4 Abeja        92.4    28 33841 United… Sourced fr… Reserve         97
##  5 Abeja        92.4    28 33847 United… This 100% … <NA>            95
##  6 Abeja        92.4    28 40428 United… Made from … <NA>            92
##  7 Abeja        92.4    28 48216 United… This new v… <NA>            93
##  8 Abeja        92.4    28 54754 United… Abeja wine… <NA>            94
##  9 Abeja        92.4    28 54763 United… Abeja's Ch… <NA>            94
## 10 Abeja        92.4    28 54891 United… Walla Wall… Estate Gro…     90
## # … with 4,055 more rows, and 9 more variables: price <dbl>,
## #   province <chr>, region_1 <chr>, region_2 <chr>, taster_name <chr>,
## #   taster_twitter_handle <chr>, title <chr>, variety <chr>, year <chr>
```