

# Tidy Data and Summarization

Jameson Watts, Ph.D.

# Agenda

1. Review (and upgrade)
2. Tame and tidy data
3. Data Summarization

# Review

# Filter, arrange, and select

Load the wine dataset and output a tibble of...

1. Pinot Noir
2. from Oregon
3. in descending order by points,
4. and ascending order by price,
5. that only shows points, price and title

# Solution

```
wine %>%
  filter(province=="Oregon") %>%
  filter(variety=="Pinot Noir") %>%
  arrange(desc(points), price) %>%
  select(points, price, title)

## # A tibble: 2,786 x 3
##   points price title
##   <dbl> <dbl> <chr>
## 1     97    65 Ken Wright 2012 Abbott Claim Vineyard Pinot Noir
## 2     96    30 Sineann 2015 TFL Pinot Noir (Willamette Valley)
## 3     96    40 Scott Paul 2009 Dix Pinot Noir (Dundee Hills)
## 4     96    60 Patricia Green Cellars 2015 Estate Vineyard Etzel Block Pi...
## 5     96    63 Ken Wright 2014 Bryce Vineyard Pinot Noir (Ribbon Ridge)
## 6     96    63 Ken Wright 2014 Abbott Claim Vineyard Pinot Noir
## 7     96    65 Ken Wright 2012 Freedom Hill Vineyard Pinot Noir (Willamet...
## 8     96    85 Alloro 2014 Estate Justina Pinot Noir (Chehalem Mountains)
## 9     96    85 The Eyrie Vineyards 2012 Original Vines Estate Pinot Noir ...
## 10    96    85 Domaine Drouhin Oregon 2011 Édition Limitée Pinot Noir (Du...
## # ... with 2,776 more rows
```

# Summarize

What are the mean price and points for Oregon Pinot Noir?

*Hint:* Use `skim()`

# Solution

```
library(skimr)
wine %>%
  filter(province=="Oregon") %>%
  filter(variety=="Pinot Noir") %>%
  arrange(desc(points), price) %>%
  select(points, price) %>%
  skim()
```

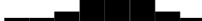

```
## Skim summary statistics
```

```
##  n obs: 2786
```

```
##  n variables: 2
```

```
##
```

```
## — Variable type:numeric —
```

##	variable	missing	complete	n	mean	sd	p0	p25	p50	p75	p100	hist
##	points	0	2786	2786	89.47	2.66	80	88	90	91	97	
##	price	7	2779	2786	44.62	20.19	9	30	42	55	275	

## Next level...

- and ('&') vs. or ('|')
- not ('!') and not equal ('!=')
- top\_n() and top\_frac()

## Use logical operators and the top\_n function to find...

1. the top 10 French or Italian wines by price.
2. showing only points, price and title
3. arranged by points descending.



# Solution

```
wine %>%
  filter(country=="France" | country=="Italy") %>%
  top_n(10,price) %>%
  arrange(desc(points)) %>%
  select(points, price, title)

## # A tibble: 10 x 3
##   points price title
##   <dbl> <dbl> <chr>
## 1    100  1500 Château Lafite Rothschild 2010 Pauillac
## 2    100  1500 Château Cheval Blanc 2010 Saint-Émilion
## 3     98  1900 Château Margaux 2009 Margaux
## 4     97  2000 Château Pétrus 2011 Pomerol
## 5     96  1200 Château Haut-Brion 2009 Pessac-Léognan
## 6     96  1300 Château Mouton Rothschild 2009 Pauillac
## 7     96  2500 Château Pétrus 2014 Pomerol
## 8     96  2500 Domaine du Comte Liger-Belair 2010 La Romanée
## 9     96  2000 Domaine du Comte Liger-Belair 2005 La Romanée
## 10    88  3300 Château les Ormes Sorbet 2013 Médoc
```

# More practice

Use logical operators and the top\_n function to find...

1. the top 5 Oregon wines by points
2. that aren't Chardonnay
3. Showing only points, price and title
4. arranged by price ascending.

# Solution

```
wine %>%
  filter(province=="Oregon") %>%
  filter(variety!="Chardonnay") %>%
  top_n(5,points) %>%
  arrange(price) %>%
  select(points, price, title)

## # A tibble: 7 x 3
##   points price title
##   <dbl> <dbl> <chr>
## 1     97    65 Ken Wright 2012 Abbott Claim Vineyard Pinot Noir
## 2     99    75 Cayuse 2009 En Chamberlin Vineyard Syrah (Walla Walla Valle...
## 3     99    75 Cayuse 2011 En Chamberlin Vineyard Syrah (Walla Walla Valle...
## 4     98    75 Cayuse 2011 En Cerise Vineyard Syrah (Walla Walla Valley (O...
## 5     97    80 Cayuse 2009 The Widowmaker Cabernet Sauvignon (Walla Walla ...
## 6     97    85 Cayuse 2009 Armada Vineyard Syrah (Walla Walla Valley (OR))
## 7     97    90 Cayuse 2011 Widowmaker En Chamberlin Vineyard Cabernet Sauv...
```

- Why are there more than 5 rows?

# Even more practice

Use logical operators and top\_frac functions to find...

1. the top 5% by points
2. of Oregon wines
3. that are neither Pinot Noir nor Chardonnay
4. Showing only points, price and title
5. arranged by points descending and price ascending.

# Solution

```
wine %>%
  filter(province=="Oregon") %>%
  filter(variety != "Pinot Noir" & variety != "Chardonnay") %>%
  top_frac(.01,points) %>%
  select(points, price, title) %>%
  arrange(desc(points), price)

## # A tibble: 26 x 3
##   points price title
##   <dbl> <dbl> <chr>
## 1     99     75 Cayuse 2009 En Chamberlin Vineyard Syrah (Walla Walla Vall...
## 2     99     75 Cayuse 2011 En Chamberlin Vineyard Syrah (Walla Walla Vall...
## 3     98     75 Cayuse 2011 En Cerise Vineyard Syrah (Walla Walla Valley (...
## 4     97     80 Cayuse 2009 The Widowmaker Cabernet Sauvignon (Walla Walla...
## 5     97     85 Cayuse 2009 Armada Vineyard Syrah (Walla Walla Valley (OR))
## 6     97     90 Cayuse 2011 Widowmaker En Chamberlin Vineyard Cabernet Sau...
## 7     96     32 Trisaetum 2016 Ribbon Ridge Estate Dry Riesling (Ribbon Ri...
## 8     96     38 Trisaetum 2015 Estates Reserve Riesling (Willamette Valley)
## 9     96     70 Cayuse 2012 Cailloux Vineyard Viognier (Walla Walla Valley...
## 10    96     75 Cayuse 2009 Camaspelo Cabernet Sauvignon-Merlot (Walla Wal...
## # ... with 16 more rows
```

# Tame and tidy data

# Philosophy (review)

- Tame data is data with understandable column names and well-formatted values
- Tidy data is data with:
  - Each variable must have its own column
  - Each observation must have its own row
  - Each value must have its own cell

The image shows three versions of a data table with columns: country, year, cases, and population. The data rows are: Afghanistan (2000, 366, 2095360), Brazil (1999, 37537, 17206362), Brazil (2000, 8488, 17404898), China (1999, 21258, 1272015272), and China (2000, 21066, 128003583).

**variables:** Vertical double-headed arrows connect the values in each column (country, year, cases, population), indicating that each column represents a single variable.

**observations:** Horizontal double-headed arrows connect the values across each row, indicating that each row represents a single observation.

**values:** Each individual data point is enclosed in a circle, highlighting that each value occupies its own unique cell in the data structure.

This is often the difference between data that is considered “long” and data that is considered “wide.”

Image credit: [https://rstudio-pubs-static.s3.amazonaws.com/396363\\_adaf67178eab4bd793bd9dd17dda70b3.html](https://rstudio-pubs-static.s3.amazonaws.com/396363_adaf67178eab4bd793bd9dd17dda70b3.html)

# Different data types

\*Each column must contain values of the **SAME** type

- Numeric (integers, fractions)
- Character (Words)
- Factor (Categories)
- Date (also includes time)
- Logical (true or false, 1 or 0)

See [here](#) for more information.



# New dataframes

So far, we've been piping operations from a single dataframe. But what if you want to save the result for later?

```
wine_oregon <- wine %>%
  filter(province=="Oregon")
```

```
wine_oregon
```

```
## # A tibble: 5,373 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     4 US      Much like ... Vintner's ...  87    65 Oregon Willame...
## 3    21 US      A sleek mi... <NA>           87    20 Oregon Oregon
## 4    35 US      As with ma... Hyland         86    50 Oregon McMinnv...
## 5    41 US      A stiff, t... <NA>           86    22 Oregon Willame...
## 6    78 US      Some rosés... Rosé of        86    25 Oregon Eola-Am...
## 7   173 US      This wine ... <NA>           91    38 Oregon Willame...
## 8   233 US      There is a... Reserve        85    28 Oregon Willame...
## 9   248 US      This seems... Estate Sin...  85    45 Oregon Willame...
## 10  251 US      Spicy and ... Papillon E...  85    22 Oregon Willame...
## # ... with 5,363 more rows, and 6 more variables: region_2 <chr>,
## #   taster_name <chr>, taster_twitter_handle <chr>, title <chr>,
## #   variety <chr>, winery <chr>
```

# Spread and Gather

These are functions to reshape your data. Let's first summarize the wine data by country and save it to a new dataframe

```
wine_country <- wine %>%
  filter(variety=="Cabernet Sauvignon" | variety=="Chardonnay" | variety=="Pinot Gris" | variety=="Syrah") %>%
  group_by(country, variety) %>%
  summarize(points = mean(points))
```

```
wine_country
```

```
## # A tibble: 90 x 3
## # Groups:   country [32]
##   country variety      points
##   <chr>    <chr>    <dbl>
## 1 Argentina Cabernet Sauvignon 86.0
## 2 Argentina Chardonnay      84.9
## 3 Argentina Pinot Gris      84.9
## 4 Argentina Syrah           85.8
## 5 Australia Cabernet Sauvignon 89.3
## 6 Australia Chardonnay      87.3
## 7 Australia Pinot Gris      87.4
## 8 Australia Syrah           91.6
## 9 Austria   Cabernet Sauvignon 87.4
## 10 Austria   Chardonnay      90.3
## # ... with 80 more rows
```

*Note:* Don't stress about the `group_by` and `summarize` functions. I'll get to that.

# Spread (from long to wide)

Now let's spread it out so that I've got one column for each variety of wine

```
wine_wide <- wine_country %>%
  spread(variety, points)
wine_wide
```

```
## # A tibble: 32 x 5
## # Groups:   country [32]
##   country `Cabernet Sauvignon` Chardonnay `Pinot Gris` Syrah
##   <chr>          <dbl>      <dbl>      <dbl> <dbl>
## 1 Argentina      86.0       84.9       84.9  85.8
## 2 Australia      89.3       87.3       87.4  91.6
## 3 Austria        87.4       90.3       90.1   89
## 4 Brazil         83        83.8        NA    NA
## 5 Bulgaria       87.8       88.5        NA    89
## 6 Canada         90        88.9       90.2  91.1
## 7 Chile          86.7       85.1        NA   88.0
## 8 Croatia        NA        NA         83    NA
## 9 England        NA        92.4        NA    NA
## 10 France        85.4       89.3       89.6  89.9
## # ... with 22 more rows
```

# Gather (from wide to long)

Then gather it back up into the original

```
wine_long <- wine_wide %>%
  gather("variety", "points", 2:5)
```

```
wine_long
```

```
## # A tibble: 128 x 3
## # Groups:   country [32]
##   country  variety      points
##   <chr>    <chr>      <dbl>
## 1 Argentina Cabernet Sauvignon  86.0
## 2 Australia Cabernet Sauvignon  89.3
## 3 Austria   Cabernet Sauvignon  87.4
## 4 Brazil    Cabernet Sauvignon   83
## 5 Bulgaria  Cabernet Sauvignon  87.8
## 6 Canada    Cabernet Sauvignon   90
## 7 Chile     Cabernet Sauvignon  86.7
## 8 Croatia   Cabernet Sauvignon   NA
## 9 England   Cabernet Sauvignon   NA
## 10 France    Cabernet Sauvignon  85.4
## # ... with 118 more rows
```

Why are there more rows than the original?

# Data summarization

# Basics

Data summarization involves

- Describing data with numerical summaries
- Visualizing data with graphical summaries

...however, there is a difference in how we describe the data depending on whether it is

- discrete, or
- continuous

# Describing discrete data

```
wine %>%
  count(country)
```

```
## # A tibble: 44 x 2
##   country      n
##   <chr>      <int>
## 1 Argentina  3800
## 2 Armenia     2
## 3 Australia  2329
## 4 Austria    3345
## 5 Bosnia and Herzegovina  2
## 6 Brazil      52
## 7 Bulgaria   141
## 8 Canada     257
## 9 Chile     4472
## 10 China       1
## # ... with 34 more rows
```

# A 'tidy' pivot table

```
wine %>%
  count(country, variety)
```

```
## # A tibble: 1,644 x 3
##   country  variety      n
##   <chr>    <chr>    <int>
## 1 Argentina Barbera      1
## 2 Argentina Bonarda    105
## 3 Argentina Bordeaux-style Red Blend    89
## 4 Argentina Bordeaux-style White Blend     1
## 5 Argentina Cabernet Blend      8
## 6 Argentina Cabernet Franc    64
## 7 Argentina Cabernet Franc-Cabernet Sauvignon    3
## 8 Argentina Cabernet Franc-Malbec      4
## 9 Argentina Cabernet Sauvignon   540
## 10 Argentina Cabernet Sauvignon-Cabernet Franc     1
## # ... with 1,634 more rows
```



## Exercise

Use filter and count to figure out which country has more Chardonnay, France or the US.

# Solution

```
wine %>%  
  filter(country=="France" | country=="US") %>%  
  filter(variety=="Chardonnay") %>%  
  count(country)
```

```
## # A tibble: 2 x 2  
##   country      n  
##   <chr>    <int>  
## 1 France    2808  
## 2 US        6801
```

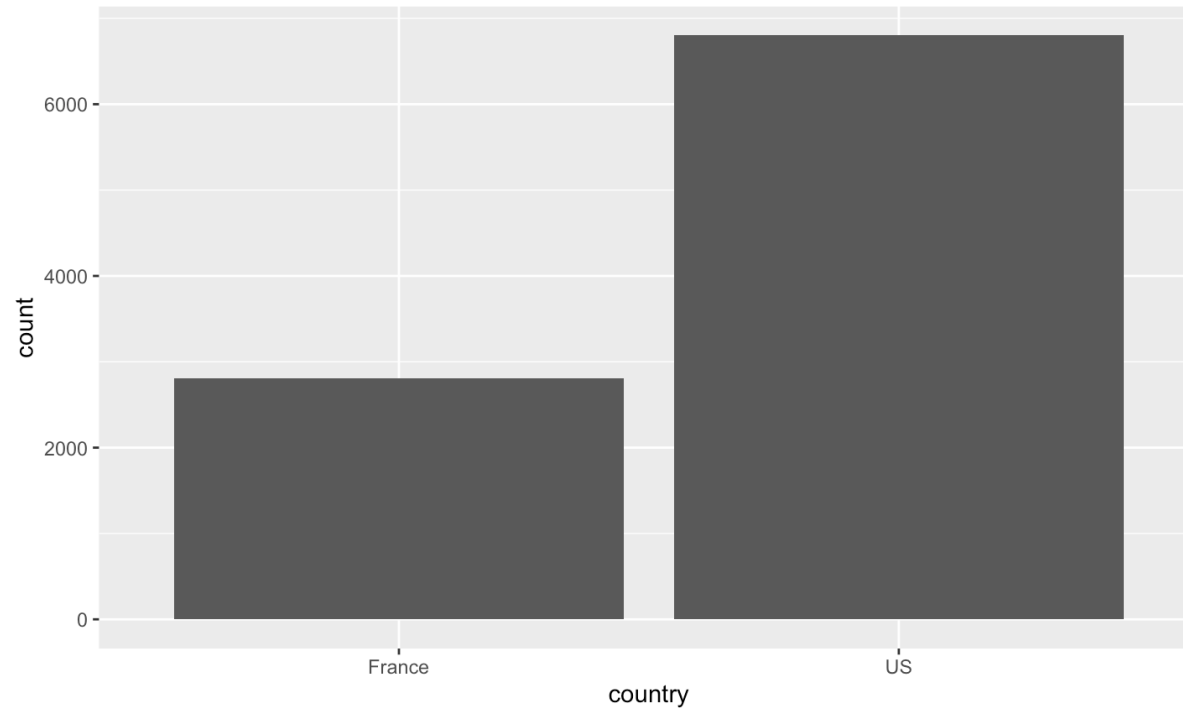
# Visualization basics

ggplot2 requires the following:

1. Data – Data to visualize.
2. Aesthetics – Mapping graphical elements to data.
3. Geometries – Or “geom,” the graphic representing the data.

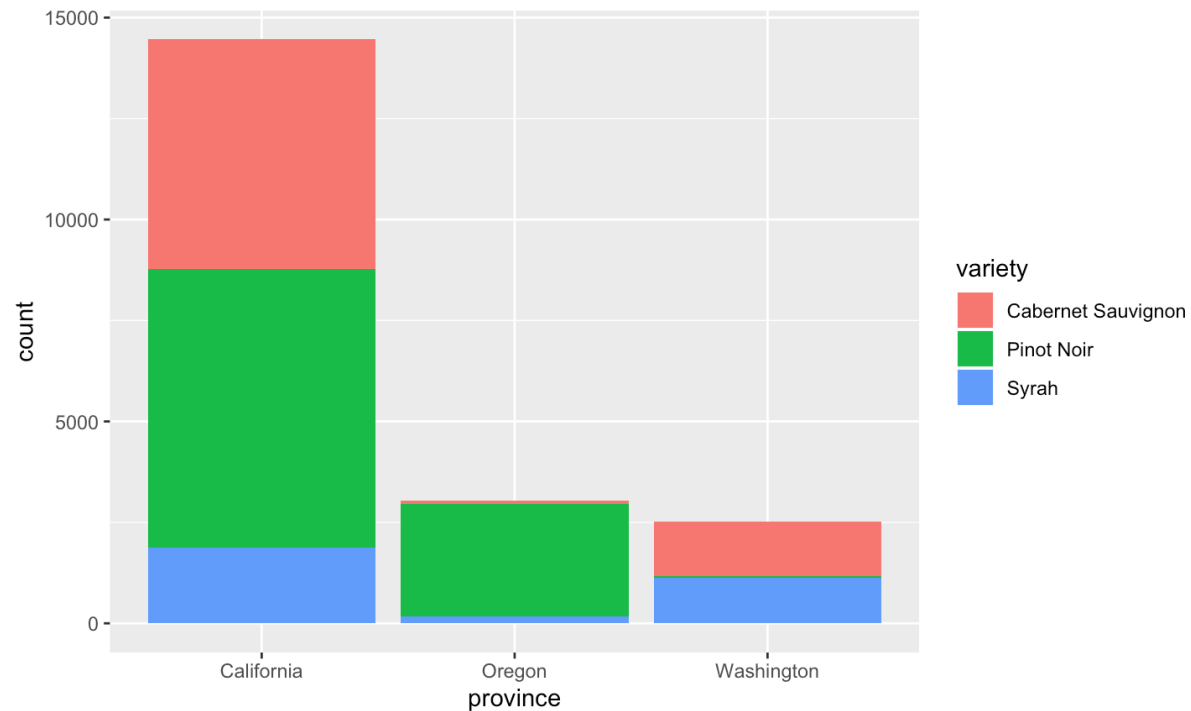
# Visualizing discrete data

```
wine %>%  
  filter(country=="France" | country=="US") %>%  
  filter(variety=="Chardonnay") %>%  
  ggplot(aes(x=country)) +  
    geom_bar()
```



# Let's try a more complicated count

```
wine %>%  
  filter(province=="Washington" | province=="Oregon" | province=="California") %>%  
  filter(variety=="Cabernet Sauvignon" | variety=="Syrah" | variety=="Pinot Noir") %>%  
  ggplot(aes(x=province, fill=variety)) +  
    geom_bar()
```



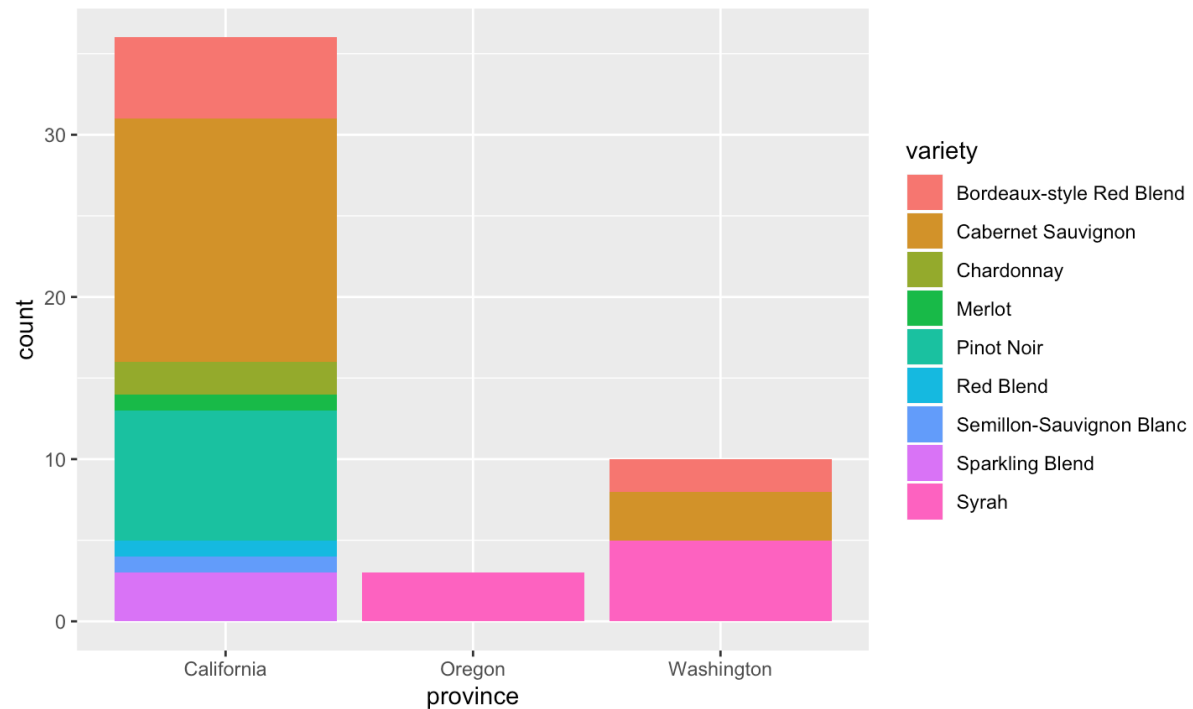
# Exercise

Create a stacked bar graph that shows

1. A count of wines
2. with greater than 97 points
3. from California, Oregon and Washington
4. stacked by variety

# Solution

```
wine %>%
  filter(points >= 98) %>%
  filter(province=="Washington" | province=="Oregon" | province=="California") %>%
  ggplot(aes(x=province, fill=variety)) +
  geom_bar()
```



# Describing continuous data

You can use the summarize function for calculating things like mean, median, variance, min/max, etc.

```
wine %>%  
  summarize(avg_points=mean(points))
```

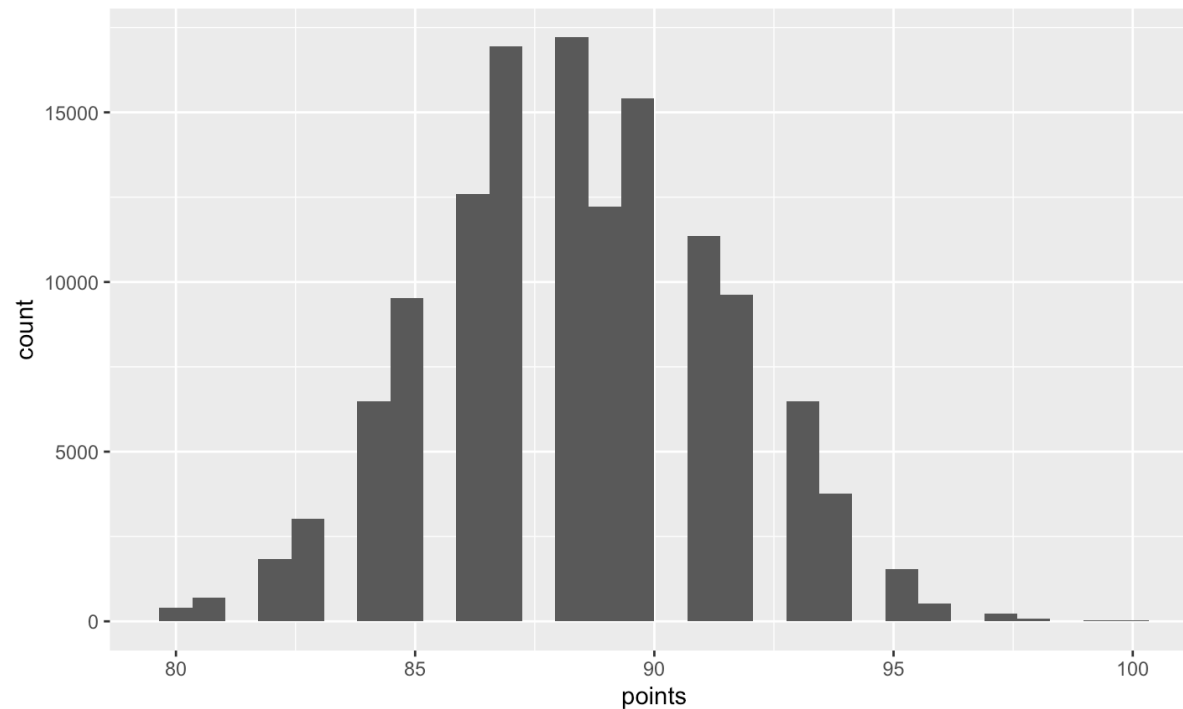
```
## # A tibble: 1 x 1  
##   avg_points  
##       <dbl>  
## 1      88.4
```



# Visualizing a continuous distribution

Of course that's not very exciting. Let's graph the distribution of points.

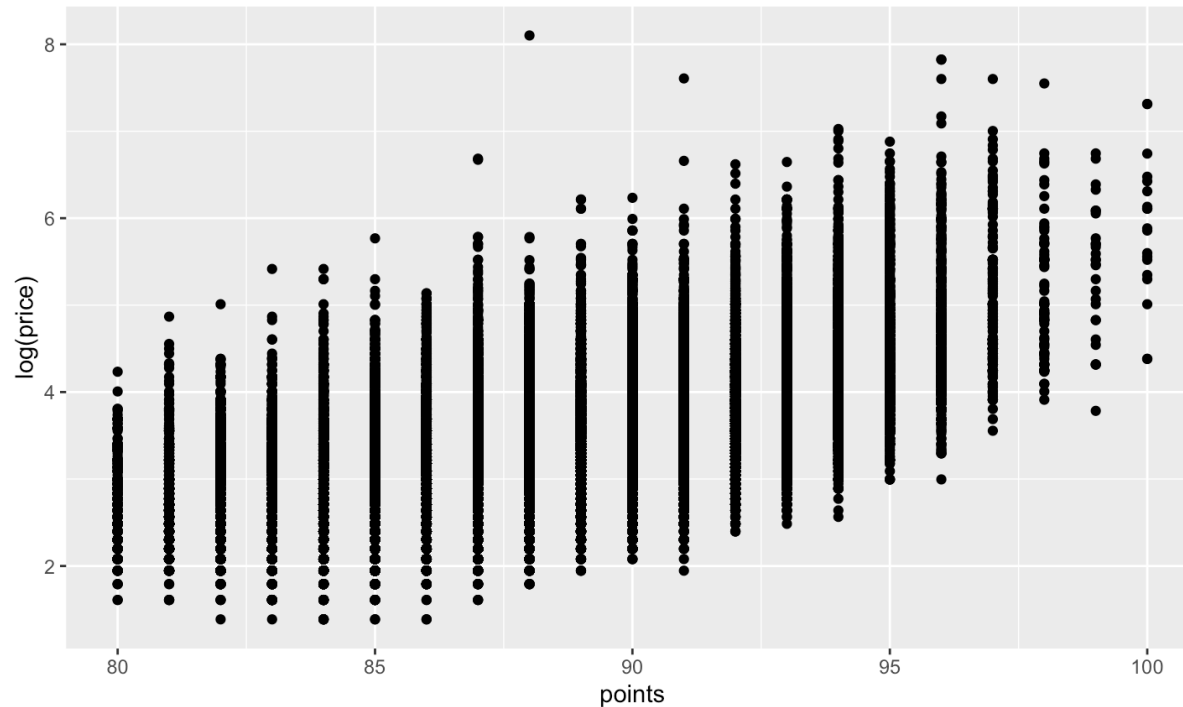
```
wine %>%  
  ggplot(aes(x=points)) +  
  geom_histogram()
```



# Visualizing two continuous variables

...or the relationship between points and price

```
wine %>%  
  ggplot(aes(x=points, y=log(price))) +  
  geom_point()
```



# Combining discrete and continuous variables

Sometimes, we want to summarize by a category

```
wine %>%
  filter(country=="US") %>%
  filter(!is.na(price)) %>%
  group_by(province) %>%
  summarize(
    count = n(),
    average_points=mean(points),
    average_price=mean(price)) %>%
  filter(count>100) %>%
  arrange(desc(average_points))
```

```
## # A tibble: 7 x 4
##   province    count average_points average_price
##   <chr>      <int>         <dbl>         <dbl>
## 1 Oregon      5359           89.1           36.5
## 2 Washington  8583           89.0           32.4
## 3 California 36104           88.6           39.0
## 4 New York    2676           87.2           22.8
## 5 Idaho        190           86.6           20.8
## 6 Michigan    111           86.2           32.4
## 7 Virginia    770           85.6           27.0
```

*Note:* the count() function used previously is just a wrapper around summarize(count=n())

# Exercise

Create a tibble that shows

1. US wines
2. grouped by province and variety,
3. summarized on count and max price
4. with a count greater than 100
5. sorted by count descending

*Hint:* don't forget to filter out the 'NA' prices

# Solution

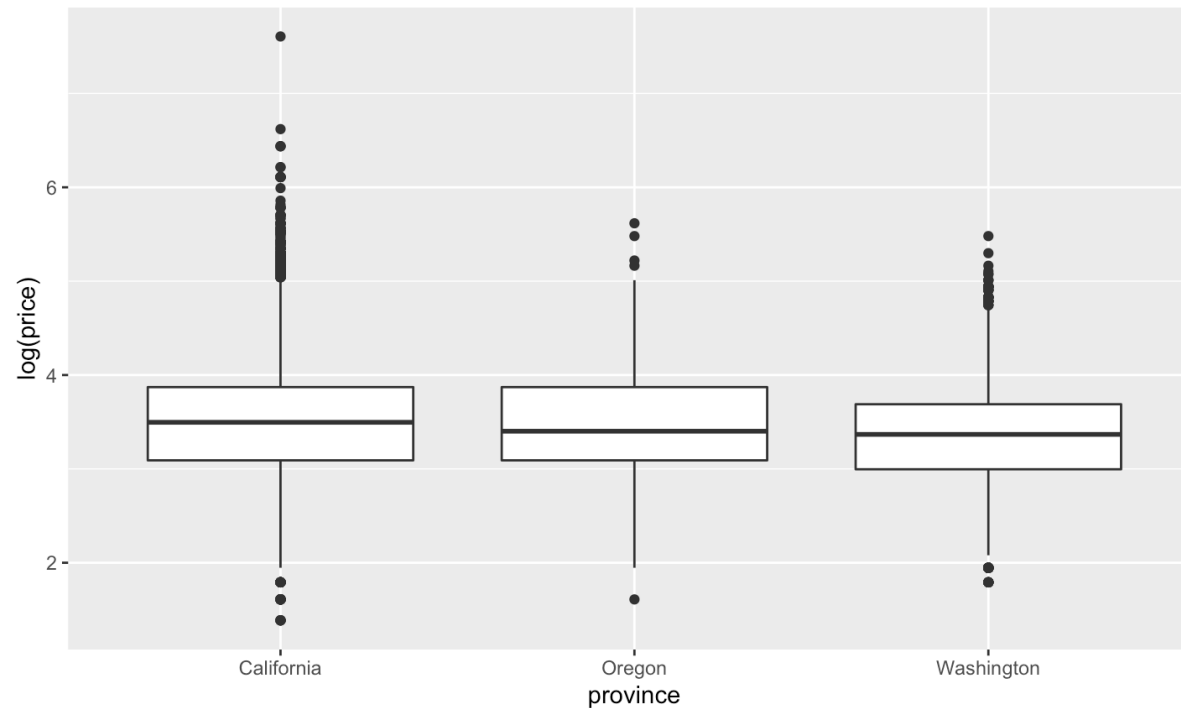
```
wine %>%
  filter(country=="US") %>%
  filter(!is.na(price)) %>%
  group_by(province, variety) %>%
  summarize(
    count = n(),
    max_price=max(price)) %>%
  filter(count>100) %>%
  arrange(desc(count))
```

```
## # A tibble: 64 x 4
## # Groups:   province [5]
##   province    variety      count max_price
##   <chr>      <chr>      <int>    <dbl>
## 1 California Pinot Noir      6875      155
## 2 California Cabernet Sauvignon 5668      625
## 3 California Chardonnay    5157    2013
## 4 Oregon     Pinot Noir      2779      275
## 5 California Zinfandel     2633      100
## 6 California Syrah        1862      750
## 7 California Sauvignon Blanc 1801       75
## 8 California Red Blend     1791      290
## 9 California Merlot        1390      200
## 10 Washington Cabernet Sauvignon 1356      160
## # ... with 54 more rows
```

# Visualizing discrete and continuous

Sometimes we want to visualize a continuous variable by category as a boxplot

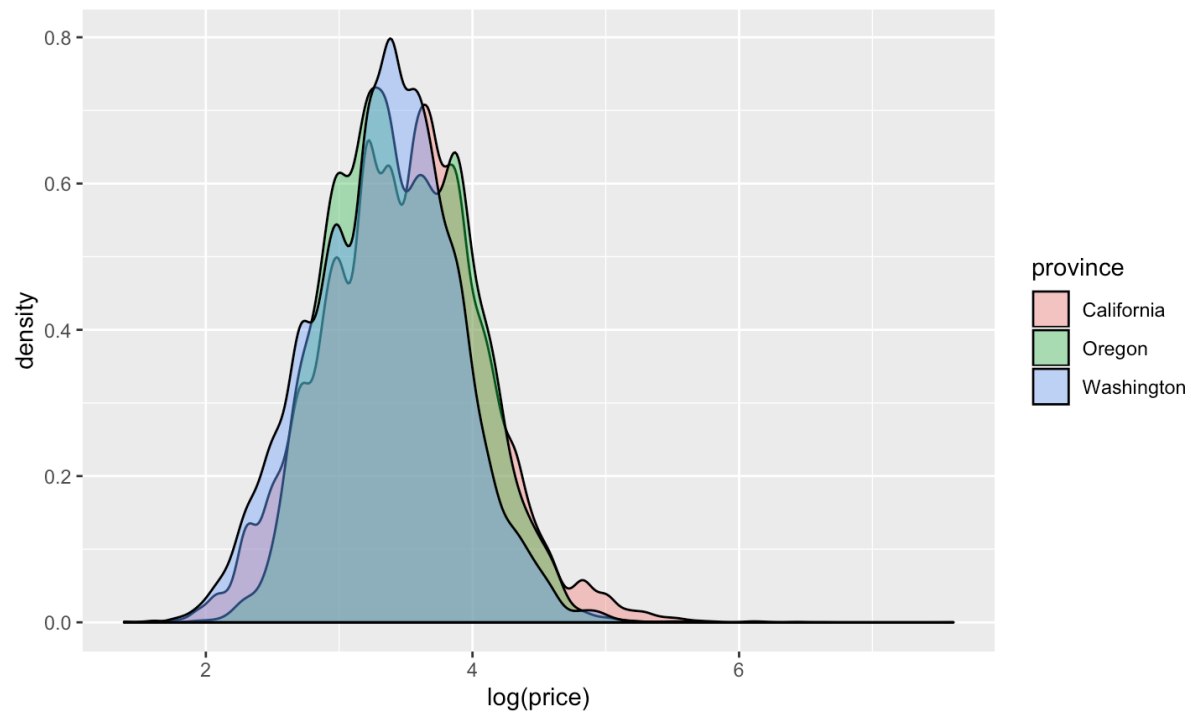
```
wine %>%  
  filter(province=="California" | province=="Oregon" | province=="Washington") %>%  
  ggplot(aes(x=province, y=log(price))) +  
    geom_boxplot()
```



# Visualizing discrete and continuous (cont'd)

...or as a density function

```
wine %>%  
  filter(province=="California" | province=="Oregon" | province=="Washington") %>%  
  ggplot(aes(x=log(price), fill=province)) +  
    geom_density(alpha = 0.4)
```



# Long exercise

Gather in groups of 3ish and...

1. Choose a driver
2. Choose a country
3. Summarize the wine data from that country (numerically and visually)
4. Write comments about what you find in the markdown
5. Make sure "echo=FALSE" on your chunks
6. Knit to HTML and email me the file.