COMPSCIX 415.2 Homework 3

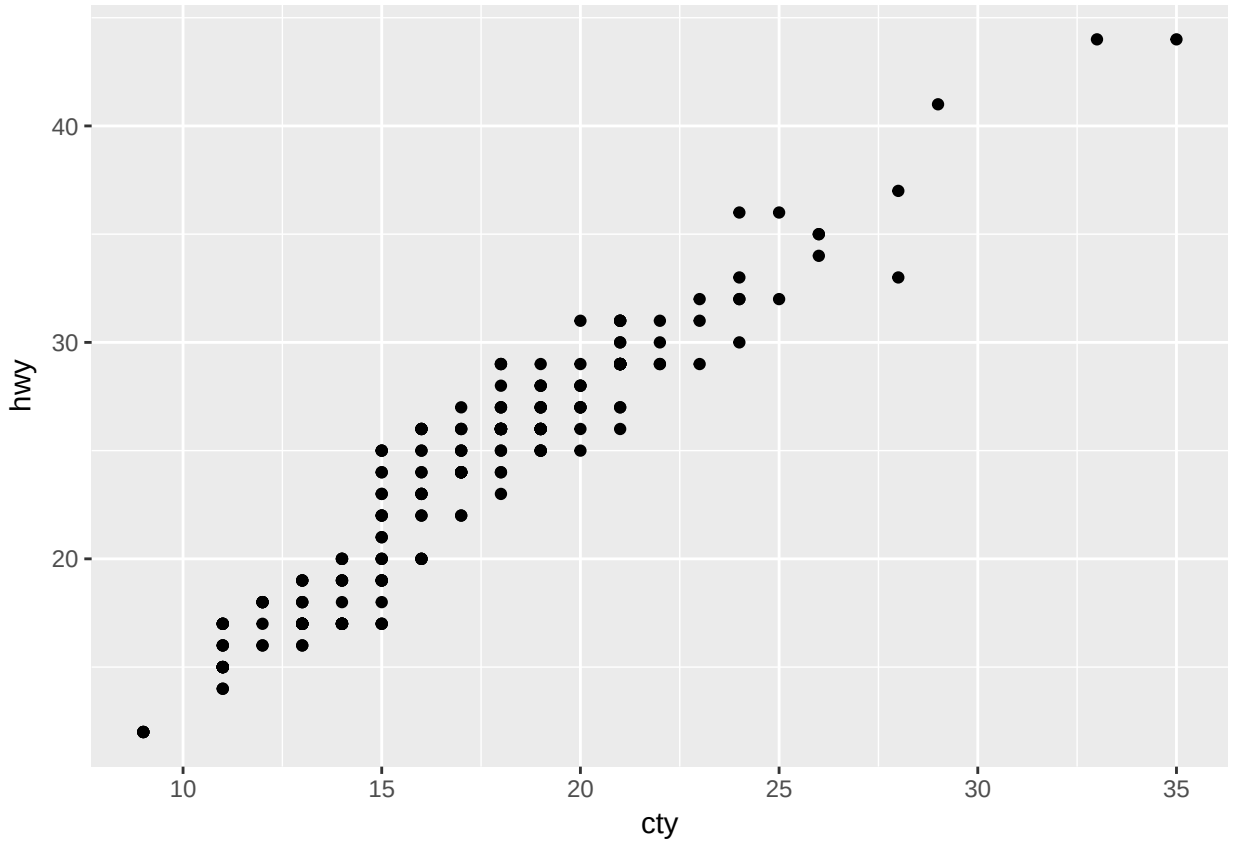
Michelle Gomez

6/24/2018

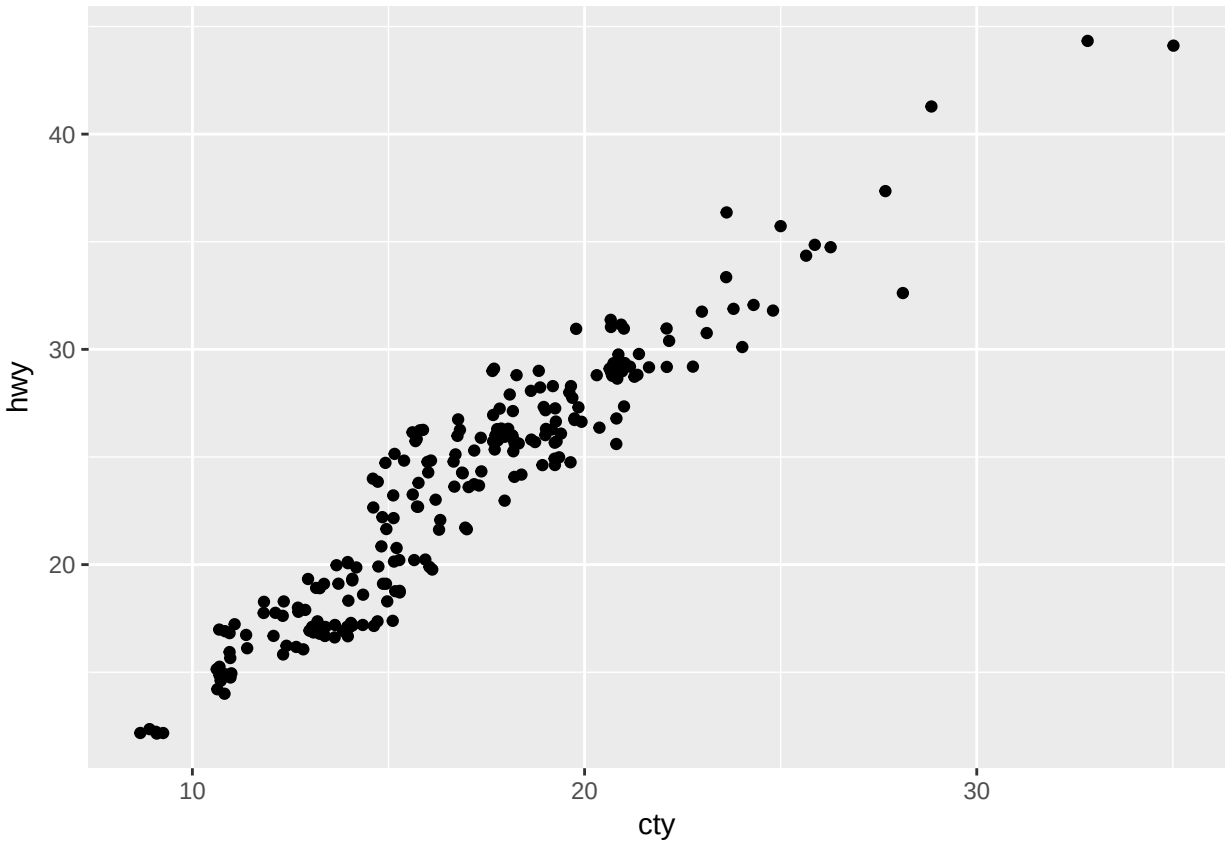
3.8.1 Exercises

1. THe graph can be improved by getting rid of the overplotting. We can plot using geom_jitter to provide a more accurate view of where the data is concentrated by adding random noise to each point.

```
ggplot(data = mpg, mapping = aes(x = cty, y = hwy)) +
  geom_point()
```




```
ggplot(data = mpg, mapping = aes(x = cty, y = hwy)) +
  geom_jitter()
```



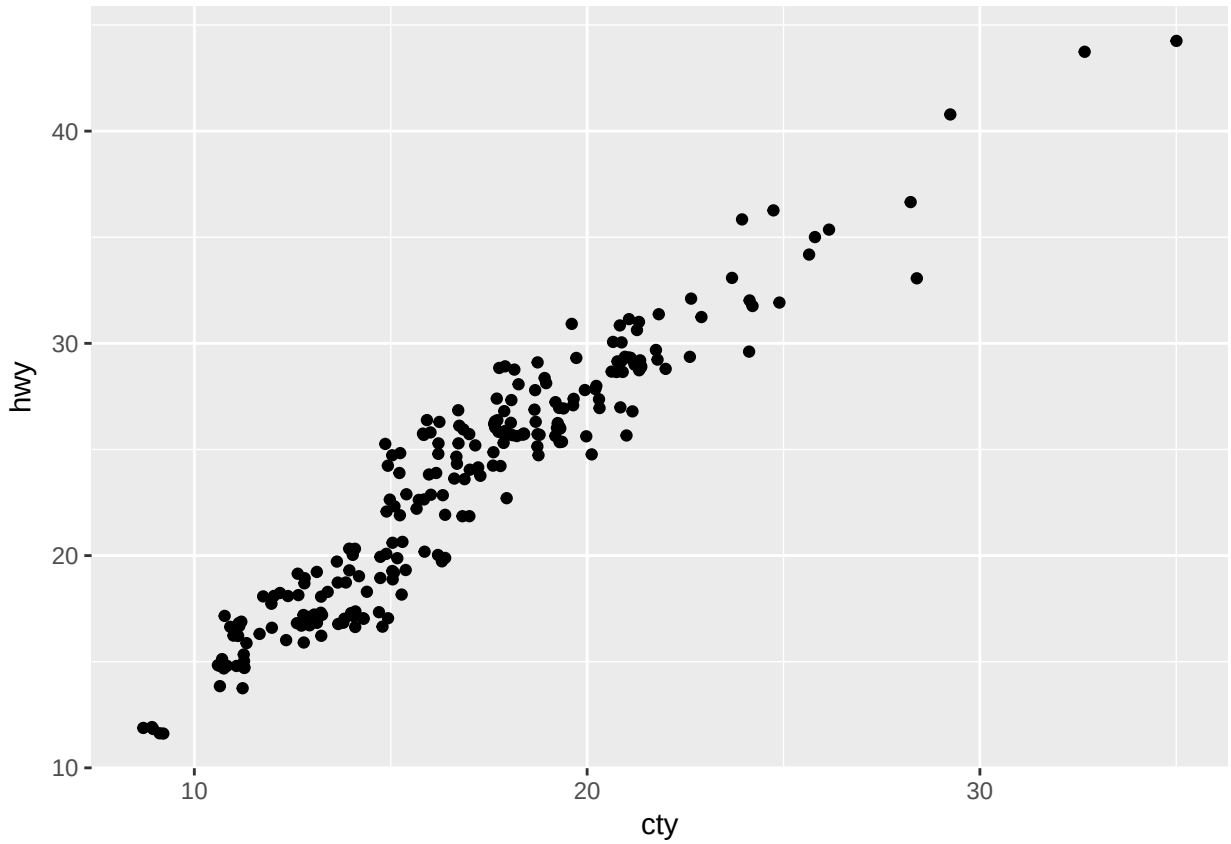
2.

According to R help:
There are two parameters to adjust the jitter– width and height.

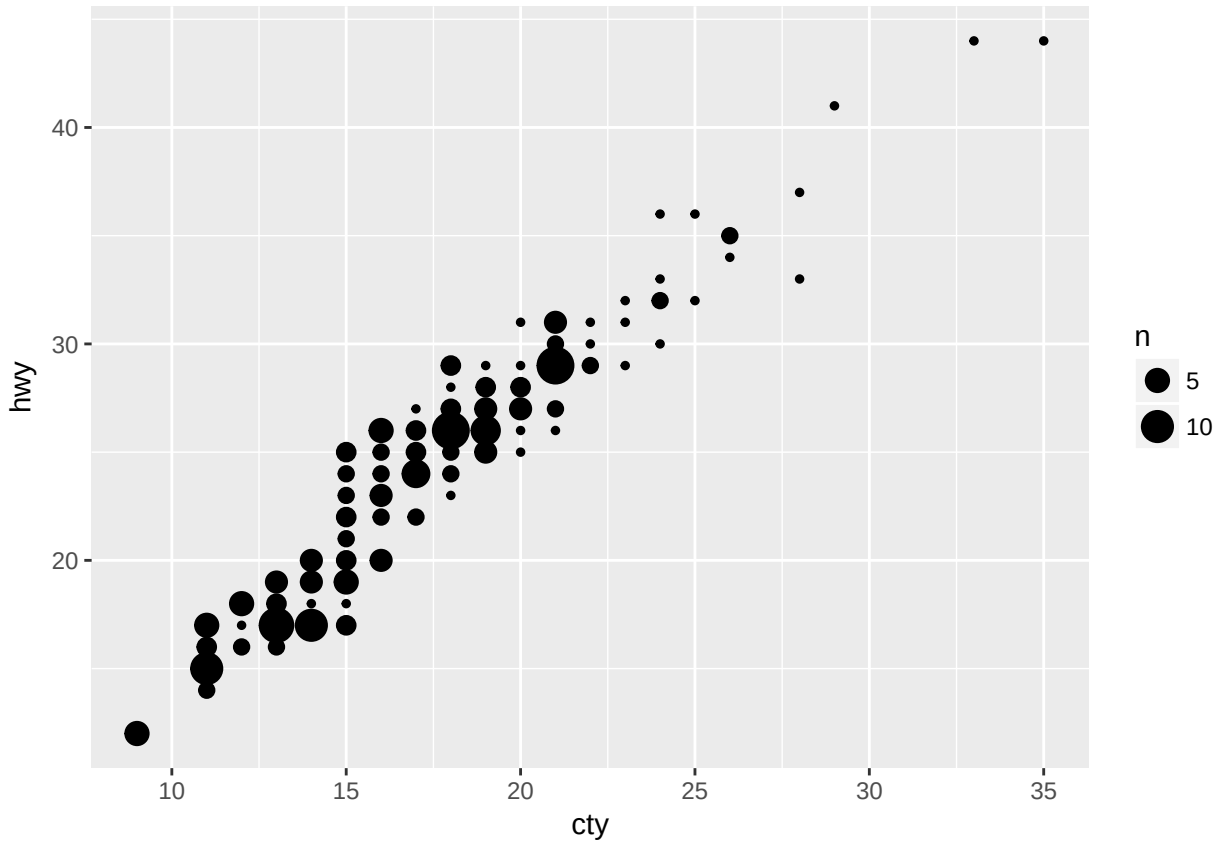
```
?position_jitter
```

3. While geom_jitter() adds random noise to each point to visualize overplotted points, geom_count() lets you see an approximate amount of observations at eahc point my mapping of scaled points.

```
ggplot(data = mpg, mapping = aes(x = cty, y = hwy)) +
  geom_jitter()
```




```
ggplot(data = mpg, mapping = aes(x = cty, y = hwy)) +
  geom_count()
```

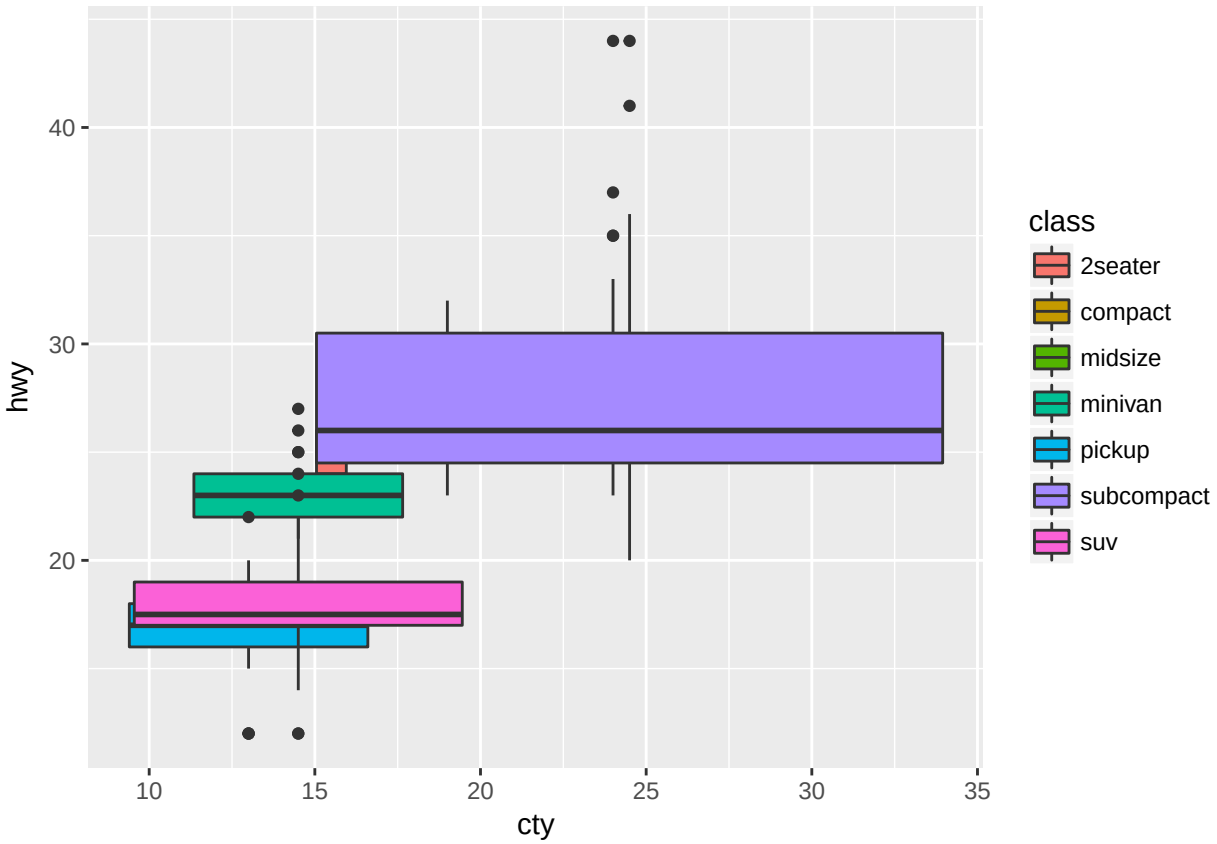



```
?geom_count()
```

4. According to the help sidebar, the default position for geom_boxplot is position_dodge. Also, in the boxplot below, the error says "position_dodge requires non-overlapping x intervals", giving us the position.

```
?geom_boxplot
ggplot(data = mpg, mapping = aes(x = cty, y = hwy, fill= class)) +
  geom_boxplot()
```

```
## Warning: position_dodge requires non-overlapping x intervals
```



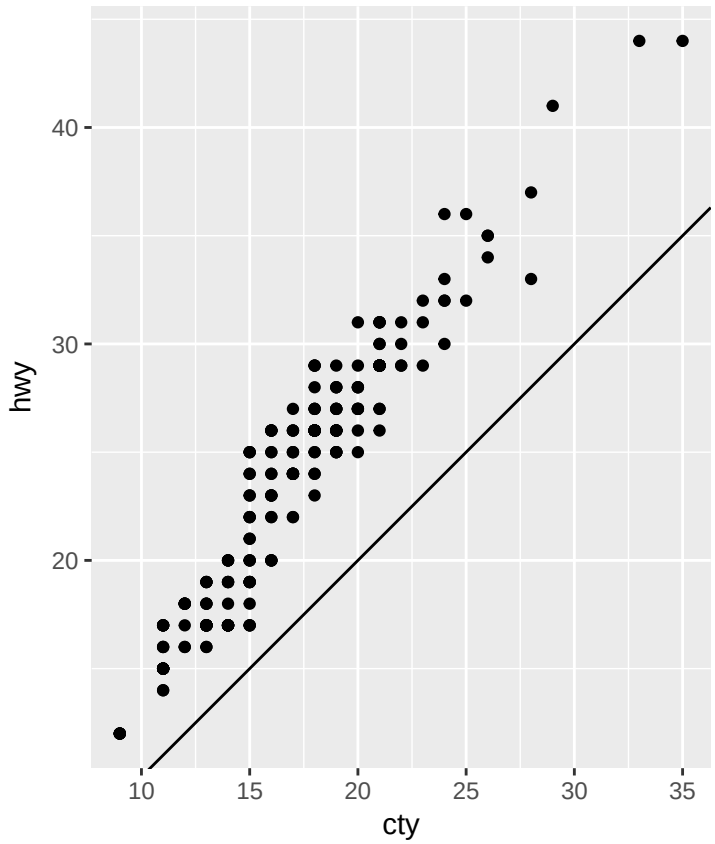
#3.9.1 Exercises

2. What does labs() do? Read the documentation. According to the documentation, labs() helps you set labels to anything from title, axis, legend, to plot names.A neat thing you can do is run all labs in one string ie. labs (x= “blah”, y= “blah”, title= “blah).

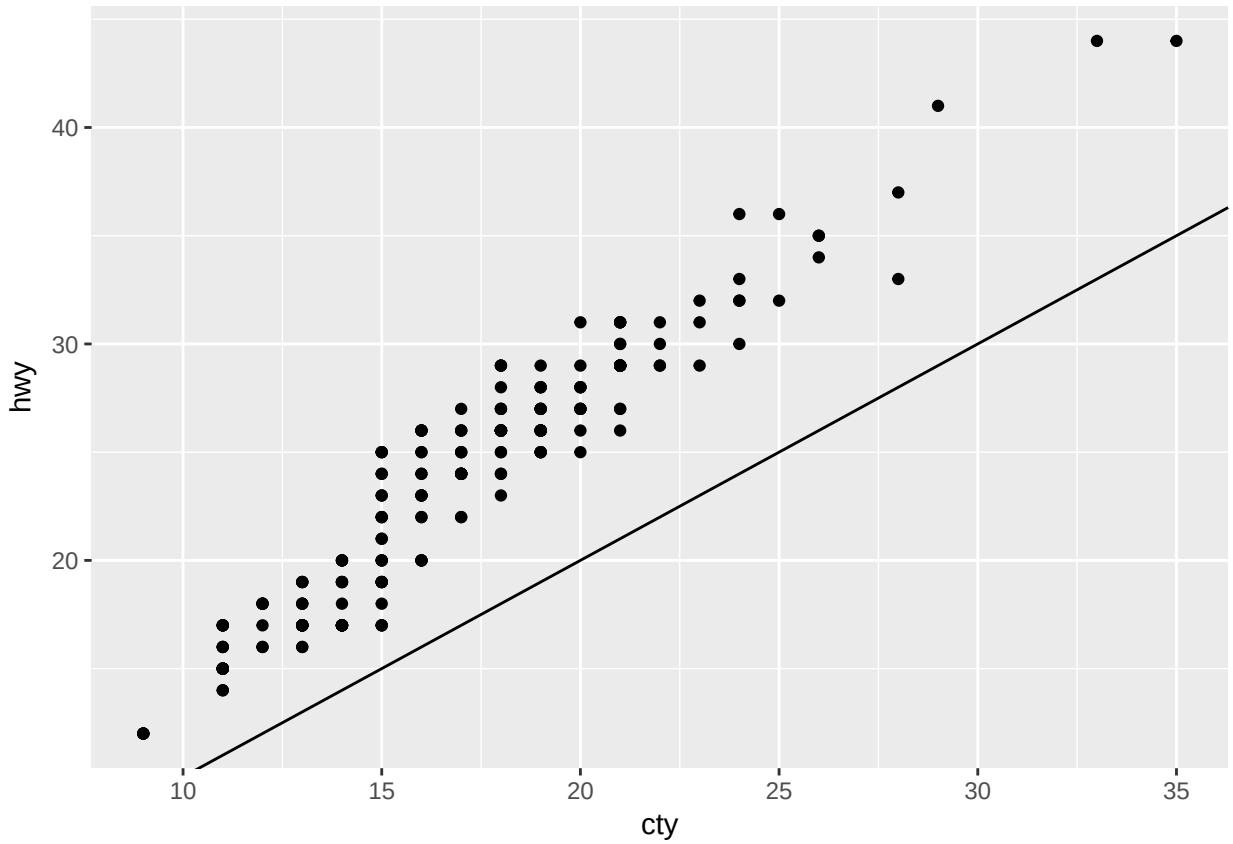
```
?labs()
```

4. The plot below tells be that city and hwy have a positive relationship.
   coord_fixed seems to change the degree of the line and may be important for visualization purposes.
   geom_abline() seems to add a reference line or rule to a plot, it’s a quick reference to see the line the data could regress to.

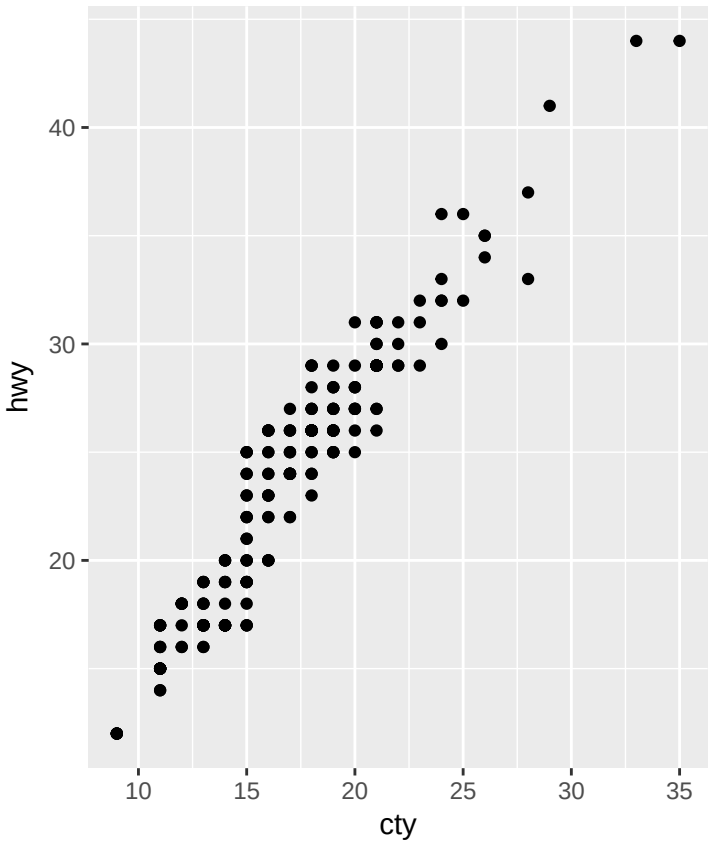
```
ggplot(data = mpg, mapping = aes(x = cty, y = hwy)) +
  geom_point() +
  geom_abline() +
  coord_fixed()
```




```
ggplot(data = mpg, mapping = aes(x = cty, y = hwy)) +
  geom_point() +
  geom_abline()
```




```
ggplot(data = mpg, mapping = aes(x = cty, y = hwy)) +
  geom_point() +
  coord_fixed()
```




```
?geom_abline()
```

Section 4.4

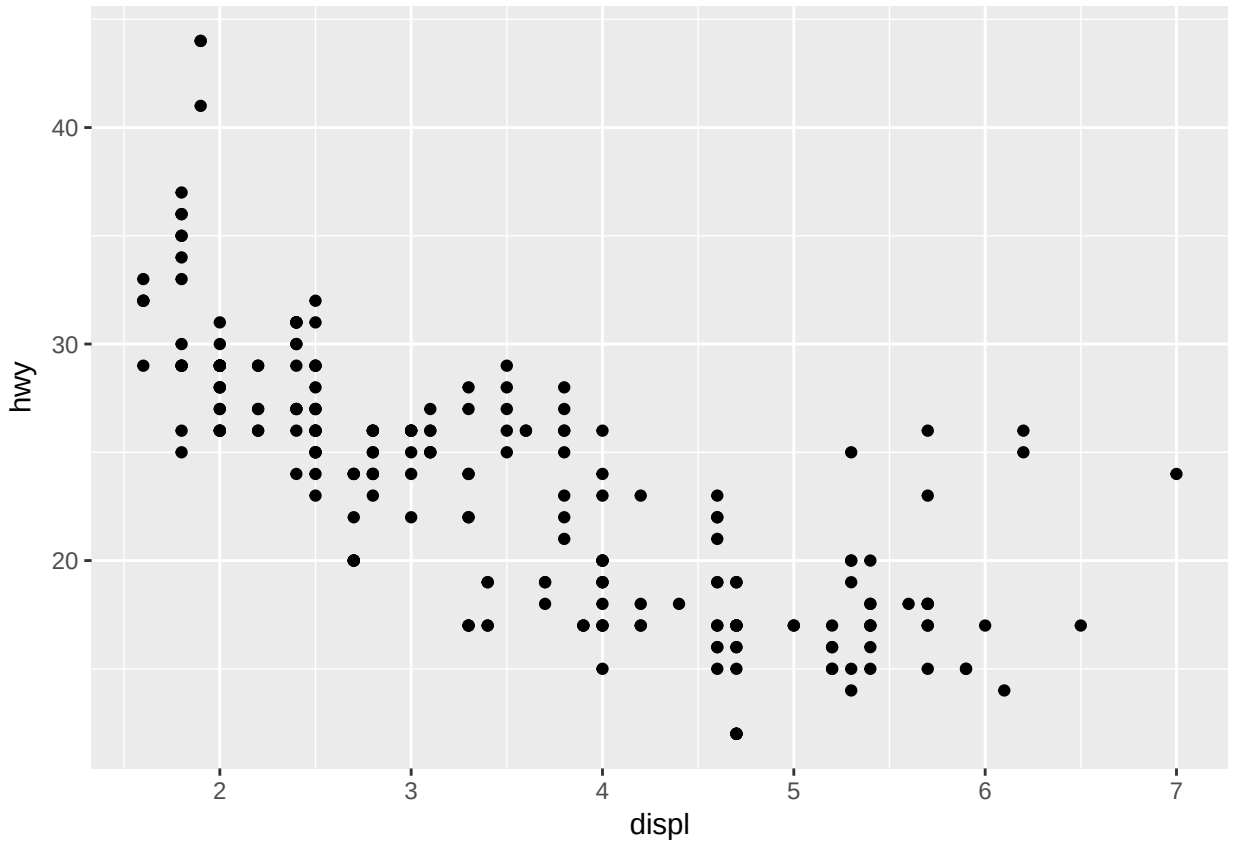
1. The "i" is dotless in my_variable, so the program doesn't understand what we are referring to.

```
#my_variable <- 10
#my_variable
```

2.

```
library(tidyverse)

ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```




```
filter(mpg, cyl == 8)
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

```
## # A tibble: 70 x 11
##    manufacturer model     displ  year   cyl trans  drv     cty   hwy fl
##    <chr>        <chr>     <dbl> <int> <int> <chr>  <chr> <int> <int> <chr>
##  1 audi         a6 quatt~  4.20  2008     8 auto(~ 4        16    23 p
##  2 chevrolet    c1500 su~  5.30  2008     8 auto(~ r        14    20 r
##  3 chevrolet    c1500 su~  5.30  2008     8 auto(~ r        11    15 e
##  4 chevrolet    c1500 su~  5.30  2008     8 auto(~ r        14    20 r
##  5 chevrolet    c1500 su~  5.70  1999     8 auto(~ r        13    17 r
##  6 chevrolet    c1500 su~  6.00  2008     8 auto(~ r        12    17 r
##  7 chevrolet    corvette   5.70  1999     8 manua~ r        16    26 p
##  8 chevrolet    corvette   5.70  1999     8 auto(~ r        15    23 p
##  9 chevrolet    corvette   6.20  2008     8 manua~ r        16    26 p
## 10 chevrolet    corvette   6.20  2008     8 auto(~ r        15    25 p
## # ... with 60 more rows, and 1 more variable: class <chr>
```

```
filter(diamonds, carat > 3)
```

```
## # A tibble: 32 x 10
##    carat cut     color clarity depth table price     x     y     z
##    <dbl> <ord>   <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
##  1  3.01 Premium I     I1       62.7  58.0  8040  9.10  8.97  5.67
##  2  3.11 Fair    J     I1       65.9  57.0  9823  9.15  9.02  5.98
##  3  3.01 Premium F     I1       62.2  56.0  9925  9.24  9.13  5.73
##  4  3.05 Premium E     I1       60.9  58.0 10453  9.26  9.25  5.66
```

```
##  5  3.02 Fair    I     I1       65.2  56.0 10577  9.11  9.02  5.91
##  6  3.01 Fair    H     I1       56.1  62.0 10761  9.54  9.38  5.31
##  7  3.65 Fair    H     I1       67.1  53.0 11668  9.53  9.48  6.38
##  8  3.24 Premium H     I1       62.1  58.0 12300  9.44  9.40  5.85
##  9  3.22 Ideal   I     I1       62.6  55.0 12545  9.49  9.42  5.92
## 10  3.50 Ideal   H     I1       62.8  57.0 12587  9.65  9.59  6.03
## # ... with 22 more rows
```

Section 5.2.4:

1.

```
library("nycflights13")
filter(flights, arr_delay > 120)
```

```
## # A tibble: 10,034 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      811            630     101       1047
##  2  2013     1     1      848           1835     853       1001
##  3  2013     1     1      957            733     144       1056
##  4  2013     1     1     1114            900     134       1447
##  5  2013     1     1     1505           1310     115       1638
##  6  2013     1     1     1525           1340     105       1831
##  7  2013     1     1     1549           1445      64.0     1912
##  8  2013     1     1     1558           1359     119       1718
##  9  2013     1     1     1732           1630      62.0     2028
## 10  2013     1     1     1803           1620     103       2008
## # ... with 10,024 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

```
filter(flights, dest %in% c("IAH", "HOU"))
```

```
## # A tibble: 9,313 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515      2.00      830
##  2  2013     1     1      533            529      4.00      850
##  3  2013     1     1      623            627   -  4.00      933
##  4  2013     1     1      728            732   -  4.00     1041
##  5  2013     1     1      739            739      0        1104
##  6  2013     1     1      908            908      0        1228
##  7  2013     1     1     1028           1026      2.00     1350
##  8  2013     1     1     1044           1045   -  1.00     1352
##  9  2013     1     1     1114            900    134        1447
## 10  2013     1     1     1205           1200      5.00     1503
## # ... with 9,303 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

```r
filter(flights, carrier %in% c("AA", "DL", "UA"))
```

```
## # A tibble: 139,504 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515      2.00      830
##  2  2013     1     1      533            529      4.00      850
##  3  2013     1     1      542            540      2.00      923
##  4  2013     1     1      554            600     -6.00      812
##  5  2013     1     1      554            558     -4.00      740
##  6  2013     1     1      558            600     -2.00      753
##  7  2013     1     1      558            600     -2.00      924
##  8  2013     1     1      558            600     -2.00      923
##  9  2013     1     1      559            600     -1.00      941
## 10  2013     1     1      559            600     -1.00      854
## # ... with 139,494 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

```r
filter(flights, month >= 7, month <= 9)
```

```
## # A tibble: 86,326 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     7     1        1           2029    212         236
##  2  2013     7     1        2           2359      3.00      344
##  3  2013     7     1       29           2245    104         151
##  4  2013     7     1       43           2130    193         322
##  5  2013     7     1       44           2150    174         300
##  6  2013     7     1       46           2051    235         304
##  7  2013     7     1       48           2001    287         308
##  8  2013     7     1       58           2155    183         335
##  9  2013     7     1      100           2146    194         327
## 10  2013     7     1      100           2245    135         337
## # ... with 86,316 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

```r
filter(flights, !is.na(dep_delay), dep_delay <= 0, arr_delay > 120)
```

```
## # A tibble: 29 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1    27     1419           1420     -1.00     1754
##  2  2013    10     7     1350           1350      0        1736
##  3  2013    10     7     1357           1359     -2.00     1858
##  4  2013    10    16      657            700     -3.00     1258
##  5  2013    11     1      658            700     -2.00     1329
##  6  2013     3    18     1844           1847     -3.00       39
##  7  2013     4    17     1635           1640     -5.00     2049
##  8  2013     4    18      558            600     -2.00     1149
##  9  2013     4    18      655            700     -5.00     1213
## 10  2013     5    22     1827           1830     -3.00     2217
```

```
## # ... with 19 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

```
filter(flights, !is.na(dep_delay),
       dep_delay >= 60, dep_delay - arr_delay > 30)
```

```
## # A tibble: 1,844 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1     2205           1720     285         46
##  2  2013     1     1     2326           2130     116        131
##  3  2013     1     3     1503           1221     162       1803
##  4  2013     1     3     1839           1700      99.0     2056
##  5  2013     1     3     1850           1745      65.0     2148
##  6  2013     1     3     1941           1759     102       2246
##  7  2013     1     3     1950           1845      65.0     2228
##  8  2013     1     3     2015           1915      60.0     2135
##  9  2013     1     3     2257           2000     177         45
## 10  2013     1     4     1917           1700     137       2135
## # ... with 1,834 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

```
filter(flights, dep_time <= 600 | dep_time == 2400)
```

```
## # A tibble: 9,373 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515      2.00      830
##  2  2013     1     1      533            529      4.00      850
##  3  2013     1     1      542            540      2.00      923
##  4  2013     1     1      544            545     -1.00     1004
##  5  2013     1     1      554            600     -6.00      812
##  6  2013     1     1      554            558     -4.00      740
##  7  2013     1     1      555            600     -5.00      913
##  8  2013     1     1      557            600     -3.00      709
##  9  2013     1     1      557            600     -3.00      838
## 10  2013     1     1      558            600     -2.00      753
## # ... with 9,363 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

3. dep_delay and arr_time are marked as NA so we can deduce that these were flights that never took off, most likely cancelled.

```
filter(flights, is.na(dep_time))
```

```
## # A tibble: 8,255 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1       NA           1630        NA       NA
##  2  2013     1     1       NA           1935        NA       NA
##  3  2013     1     1       NA           1500        NA       NA
```

```
##  4  2013     1     1       NA            600        NA       NA
##  5  2013     1     2       NA           1540        NA       NA
##  6  2013     1     2       NA           1620        NA       NA
##  7  2013     1     2       NA           1355        NA       NA
##  8  2013     1     2       NA           1420        NA       NA
##  9  2013     1     2       NA           1321        NA       NA
## 10  2013     1     2       NA           1545        NA       NA
## # ... with 8,245 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

4. Any value to the power of 0 is equal to 1, hence why NA ^ 0 is not missing. The expression NA | TRUE, is not missing because the missing value's false/true status doesn't matter.FALSE & "value"will always be false. NA * 0 is NA because multiplying anything times something missing is a missing value.

```
NA * 1
```

```
## [1] NA
```

5.4.1 Exercises

```
select(flights, dep_time, dep_delay, arr_time, arr_delay)
```

```
## # A tibble: 336,776 x 4
##    dep_time dep_delay arr_time arr_delay
##       <int>     <dbl>    <int>     <dbl>
##  1      517      2.00      830     11.0
##  2      533      4.00      850     20.0
##  3      542      2.00      923     33.0
##  4      544     -1.00     1004    -18.0
##  5      554     -6.00      812    -25.0
##  6      554     -4.00      740     12.0
##  7      555     -5.00      913     19.0
##  8      557     -3.00      709    -14.0
##  9      557     -3.00      838    - 8.00
## 10      558     -2.00      753      8.00
## # ... with 336,766 more rows
```