Exploratory Data Analysis

Lesson 5 Notes

Explore Many Variables

Multivariate Data

In the last lesson, you learned how to examine the relationship between two quantitative variables. In this lesson, you'll learn how to examine three or more variables at a time. Let's get started by hearing from Moira about how she added one more variable to her scatter plot of perceived audience size.

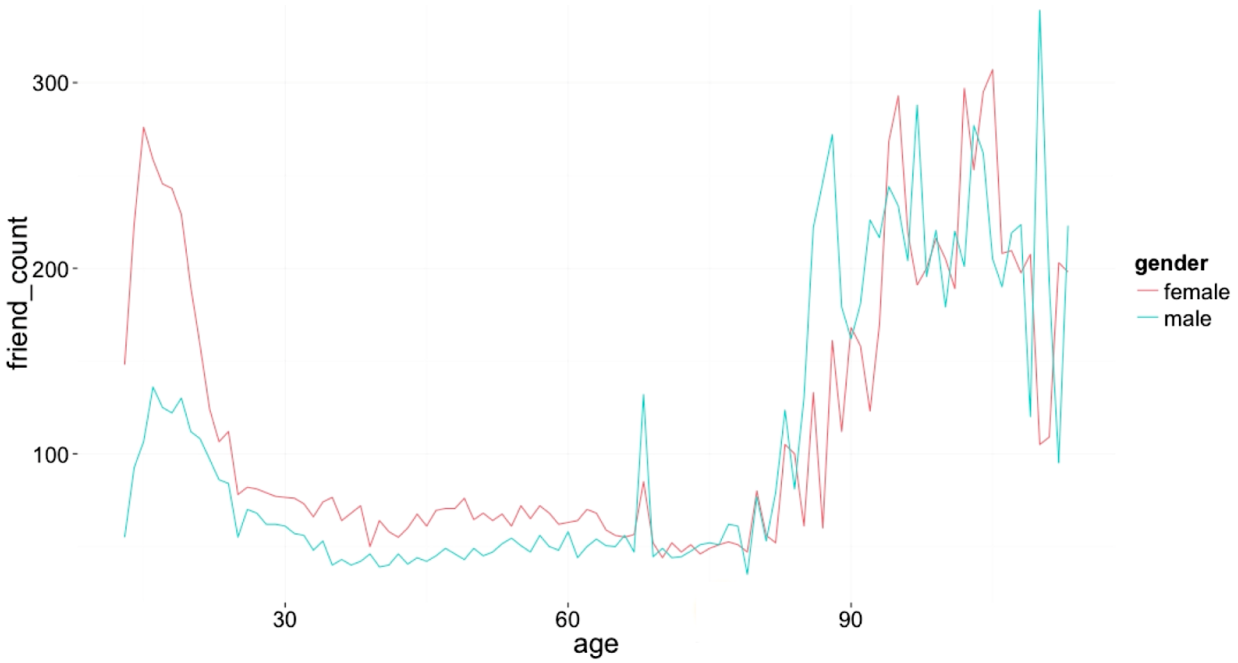
Perceived Audience Size by Age

My next question was, well, okay, so people aren't very good at guessing their audience sizes. But maybe, maybe people who are older, maybe they have a better sense, than teenagers, for example. So the next plot that we did was another scatter plot, but this time we added a third level, where we added color to represent the age of the survey respondent. And you can see again we have this horizontal stripes for people who are guessing that there are 50 or 100 people in their audience. But I don't see any pattern in the color. I think this is actually kind of a dead end. This is one example of kind of a failure. I can tell if younger people were more accurate than older people, there is too much over plotting. In this plot there are too many dots on top of each other and color really doesn't add much.

Programming Quiz: Third Qualitative Variable

It looks like Moira didn't really get anywhere by adding age as color to her plot. But I want you to know that's okay. Sometimes when we conduct exploratory data analysis we do reach dead ends. Let's see if we can get any further in examining our relationship between friends count and age by adding a third variable. Previously we noted that female users have more friends on average than male users. And, we might wonder, is this just because female users have a different age distribution? Or, maybe conditional on age, the differences are actually larger.

Copyright © 2014 Udacity, Inc. All Rights Reserved.



Here's a box plot of ages by gender. Now, I'm going to add the mean for each gender to the box plots, using stat summary. Here we can see the averages marked by an x since I used shape=4. Since male users are a bit younger, we might actually think a simple male to female comparison doesn't capture their substantial differences in friend count. Let's look at median friend count by age and gender instead. A lot of this code should look familiar, and if it doesn't, I want you to post any questions that you have about it in a discussion. When I run this code, we can see that nearly everywhere the median friend count is larger for women than it is for men. Now there are some exceptions, and this includes these noisy estimates for our very old users. Now, I'm using old with quotation marks here, since we're not really confident about these reported ages. And notice that users reporting to be of reported gender. You're going to reproduce the same plot, but first let's see if you can create the summary data to create it. Recall that we can produce the same summary data underlying this plot by using the dplyr package. We can divide the data by age and gender and then compute the median and mean friend count for each sub-group. In this next program assignment you're going to do just that by using the group buy, summarize, and arrange functions from the dplyr package.

Answer:

For this question, you need to create the data frame that would give us the data to construct this plot. Here's how I would go about constructing the code. First, I'll load the dplyr package and then I'm just going to leave myself a comment that I'm going to chain these functions together. So I'm using this symbol. I'm going to save my data frame as pf.fc by age gender, and I'm going to work from my existing data frame and group it. So, here's my first chain command, and I'm going to group by age and gender. Now, I'm going to summarize getting the mean friend count, the median friend count, and n, the number of people in each group. Now, summarize will remove one layer of grouping when

Copyright © 2014 Udacity, Inc. All Rights Reserved.

it runs, so we'll remove the gender layer. So, we need to run ungroup one more time to remove the age layer and finally I'll arrange my data frame by age. Now, it looks like that I have everything, but I actually forgot to filter or subset the data. I could use the subset command but I'm actually going to show you the filter command. I'll filter the data so that I remove any people that have a gender marked as NA and then, I just need to remember to chain that together with the rest of the functions. Alright, let's run this code and see if our data frame looks good. It looks like I actually forgot to run my library first, so let me do that. We have our new data frame up here, so let's print out some of the rows to make sure we're right. I'll print out a couple of the first rows to the console, and I can see that I have my groups split by age and gender, the mean friend count, the median friend count and n, the number of groups.

Programming Quiz: Plotting Conditional Summaries

Now that you've got this data, I want you to construct this plot that shows the median friend count for each gender as age increases. Now, the plot should be identical to this one that we created before. You code just will look slightly different, and you want to make use of this data frame.

Answer:

To create this plot, we're going to pass in our variables to ggplot as usual. We'll set x equal to age, and y equal to median friend count. And we'll set data equal to the data frame that we just created. So pf.fc by age, gender. Now I just need to add gm line and pass up the color parameter inside the aesthetic wrapper. This time the color will take on the value of gender. This will give me one line for males and one line for females. Running this code, we get the same exact plot. Hopefully you're feeling like a data guru at this point.

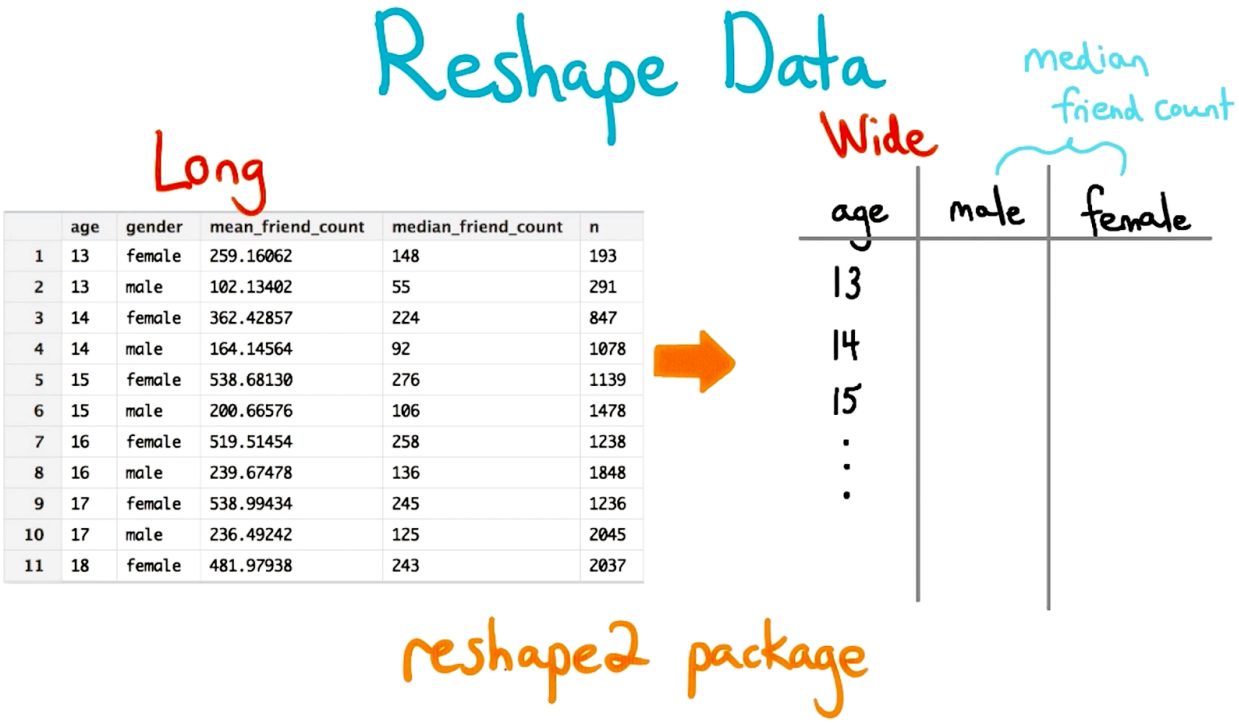
Thinking in Ratios

This can be useful if we want to inspect these values, or carry out further operations to help us understand how the difference between male and female users varies with age. For example, looking at this plot, it seems like the gender difference is largest for our young users. It would be to put this in relative terms though. So, let's answer a different question. Let's answer the question, how many times more friends does the average female user have than the male user? Maybe, females have twice as many friends as male users, or maybe it's ten times as many friends.

Wide and Long Format

Copyright © 2014 Udacity, Inc. All Rights Reserved.

To answer that question, we need to rearrange our data a little bit. Right now, our data is in long format. We have many rows. And, notice how that the variables that we grouped over, male and female, have been repeated. They're repeated for each year. So let's do something else besides this long data format. What we're going to do is convert it to a wide format. This new data frame will have one row for each age, and then we'll put the median friend count inside of males and females. Many times when computing with and exploring data, it's helpful to move back and forth between these different arrangements. To carry this out in R, we're going to be using the reshape2 package. Now, if the difference between wide and long data is still a little bit fuzzy to you, I recommend pausing the video right now. You can read through another example in the instructor notes before moving on.



	age	gender	mean_friend_count	median_friend_count	n
1	13	female	259.16062	148	193
2	13	male	102.13402	55	291
3	14	female	362.42857	224	847
4	14	male	164.14564	92	1078
5	15	female	538.68130	276	1139
6	15	male	200.66576	106	1478
7	16	female	519.51454	258	1238
8	16	male	239.67478	136	1848
9	17	female	538.99434	245	1236
10	17	male	236.49242	125	2045
11	18	female	481.97938	243	2037

Reshaping Data

If you're ready, let's go through the code to reshape our data frame into a new one. First, let's install and load the R reshape two package. Now, let's create a variable for a new data frame that will be in wide format. I'll use the same variable name for the data frame and just add wide to the end. Now, we're going to make use of the dcast function, which comes with the R shape two package. The letter d is used since we want the result to be a data frame. If we wanted an array or a matrix we could use a cast. We specify the data set we are going to change and then we put in a formula. So, here's the data frame I want to modify and then this is where I'll enter my formula. Now, the first part of the formula, or the part to the left of the tilde sign, will list the variables I want to keep with an addition sign in between them. Here I just want to keep h. On the right side of the tilde, we use the gender variable since we want male and female users to have their own columns for median friend count in the data frame. And finally, we set value dot var equal to median friend count because value dot var holds the

Copyright © 2014 Udacity, Inc. All Rights Reserved.

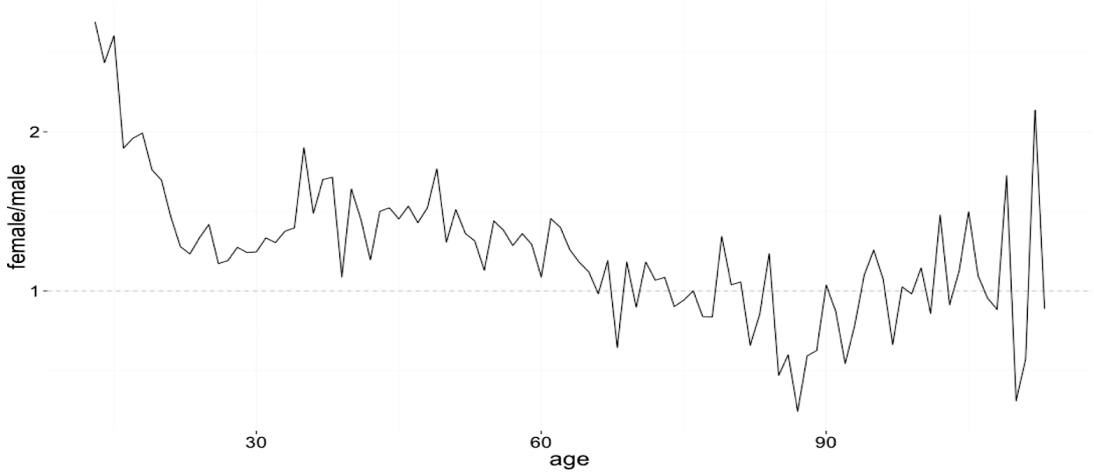
key measurements or their values in our new data frame. And it looks like that I forgot quotes around this variable.

So let me add that. Now, we should get a new data frame. And let's look at some of these data frames, so that way, we can make sure we understand this dcast function. Here are some of the rows printed out and notice I have my age column, my female median friend count and my male median friend count. See if you can recreate these steps on your own and try playing around with the dcast and melt functions, and the reshape to package. The melt function will allow you to convert the wide data back to the original long format.

Programming Quiz: Ratio Plot

We've got our new data frame so let's make use of it. I want you to plot the ratio of females to males to determine how many times more friends the average female user has, compared to the number of friends the average male user has. You'll also add some cool features to the plot, but I'll describe that in the programming exercise.

Answer:



ights Reserved.

Now, this plot might have been a little bit tricky, so let's walk through it together. I'll assign age to the X parameter, and then I'll assign females divided by males, to the Y parameter. This will give me my ratio. And then I just need to make sure that I pass my newest data frame to data. We'll add a geome line to connect the points, and then we'll also add a horizontal line. This geom_hline will take a couple parameters. We'll set the y-intercept to one, the alpha equal to 0.3, and the line type equal to two. Running this code, we get out ratio plot. We can easily see that for very young users, the median female user has over two and a half times as many friends as the median male user. Clearly, it was helpful to condition on age in understanding the relationship of gender with friend count. This helped assure us this pattern is robust for users of many different ages. And it also highlighted where this difference is most striking. Now, there are many processes that can produce this difference, including the biased distribution from which this pseudo Facebook data was generated. One idea which shows the complexity of interpretation here, is that people from particular countries who more recently joined Facebook are more likely to be male with lower friend counts.

Programming Quiz: Third Quantitative Variable

In the previous examples, we were looking at our data Age and Friend Count across the categorical variable Gender. Usually, color or shape tend to be aesthetics for representing such changes over a categorical variable. But what if we looked at Age and Friend Count over, say, another numerical variable? For example we might notice that since users are likely to accumulate friends over time using Facebook that Facebook tenure is important for predicting friend count. Tenure or how many days since registering with Facebook is associated with age. The first people to start using Facebook were college students as of 2004 and 2005. I was lucky enough to be part of that group and joined the site on February and have over 1,200 of them. On the other hand, 14 year-old users, have had less time to accumulate the same number of friends. One way to explore all four variables friend count, age, gender and tenure is using a two-dimensional display like a scatter plot. And we can bend one of the quantitative variables and compare those bends. In this case, we can group users by the year that they joined. So let's create a new variable called year_joined in our data frame. This variable is going to hold the year that our users first joined Facebook. In our next program and exercise, you're going to create this variable, year_joined, and put it inside the data frame. You need to make use of the variable tenure and use 2014 as the reference year.

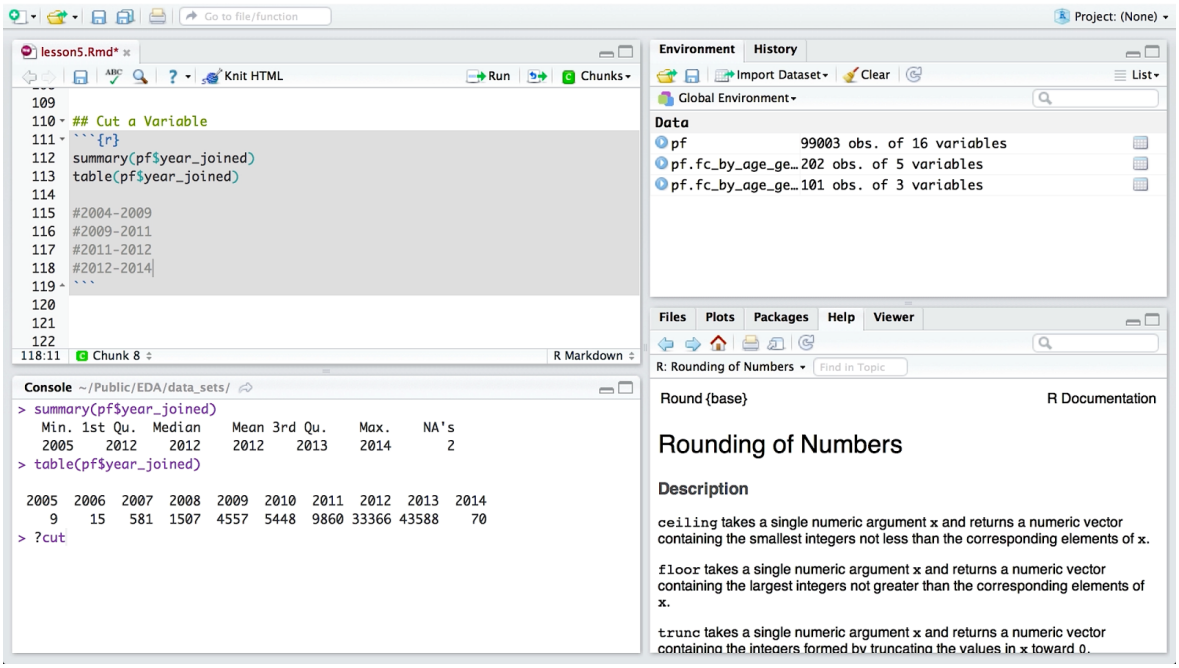
Answer:

Copyright © 2014 Udacity, Inc. All Rights Reserved.

For this programming assignment, you need to create the variable year_joined, and assign it to the data frame pf. To do that, we just need to take our reference year 2014, and subtract off our tenure, or the number of days someone's been active on Facebook. Now, tenure is measured in days rather than in years, so we need to divide this by 365. I also need to make sure that I actually access the variable so I need to include pf right here with a dollar sign. Now, this number will give me a year with a little bit of extra if there's a decimal. Now, I don't really care about the decimal, since that doesn't make a full year, so I just want to floor this number. The function floor will return the greatest integer that's not larger than this number. When I run the code, it doesn't look like much happened. But if I check my data frame, I can see that I have another variable. There it is, year_joined.

Programming Quiz: Cut a Variable

We've got our new variable your joined so let's look at its summary it looks like most of our users joined in 2012 or 2013 and since the values for this variable are discreet and the range is pretty narrow I'm going to go ahead and table this variable as well. Here we can see the distributions of users and each year joined. Notice that there isn't much data here about early joiners. To increase the data we have in each tenure category, we can group some of these years together. The cut function is often quite useful for making discrete variables from continuous or numerical ones, sometimes in combination with the function quantile. Now what I want you to do is to look at the documentation for cut, and refer to the link in the instructor notes to complete this next programming exercise. Your task is to cut the variable year joined to create four bins, or buckets of users. The bins will be from 2004 to 2009, 2009 to 2011, 2011 to 2012, and then 2012 to 2014.



Copyright © 2014 Udacity, Inc. All Rights Reserved.

Answer:

For this programming exercise, you need to create a variable called, year joined bucket. That bin together users depending on which year they joined Facebook. So, for example, this would be one bucket, this would be one bucket, this would be a bucket, and then these users would be another bucket. To do this, we just need to use the cut function on the variable year joined. I just need to tell cut on what years I should split my data. So I'm going to split at 2004, 2009, 2011, 2012, and 2014. Running this code, I can see that I got a new variable inside of my data frame.

Programming Quiz: Plotting It All Together

We've done two things up to this point. We created a variable called year_joined, based on the tenure variable, and we converted year_joined, to the variable year_joined_bucket. A categorical variable. That bin their users into different groups. Let's table this new variable to see the distribution in each group. Here we can see that we have our four bins of users, depending on when they joined Facebook, and it looks like two people have a value of NA. Let's use this new year joined bucket variable to create a line graph Like we did for gender at the start of the lesson. As a reminder, here's the code that generated this plot earlier. Also, notice how we compute the median friend count for each age using the fun.wide parameter, and the summary for the stat parameter. Using a similar code structure that we see here. I want you to create a plot for friend count versus age, so that each year join bucket has its own line on the graph. In other words, each bucket will be a color line that tracks the median friend count across the age of users, just as it did in this plot for genders.

Answer:

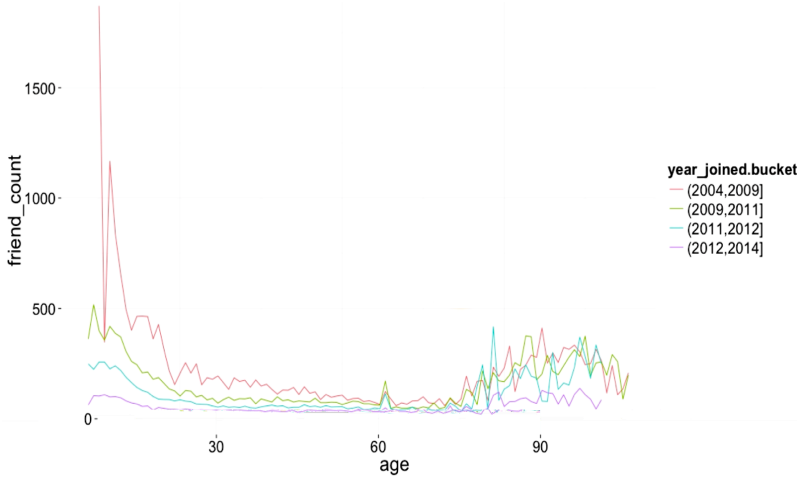
To create this new plot, we really just need to switch out gender with our year_joined.bucket variable. This is the new categorical variable that will take the place of color. Now, I'm also using year_joined.bucket here so that way, I can exclude the two people who have values of NA. I'll clean up my code just a little bit, and then I'll run it to create my plot. And now, here's the plot that examines the relationship between friend_count and age, split up by year_joined.bucket variable.

Programming Quiz: Plot the Grand Mean

Looking at this plot, we can see that our suspicion is confirmed. Users with a longer tenure tend to have higher friend counts, with
the exception of our older
users, say, about 80 and up. To
put these cohort specific
medians in perspective, we can

Copyright © 2014 Udacity, Inc. All Rights Reserved.

change them to cohort specific means. And then plot the grand mean down here as well. The grand mean is the line we saw in lesson four when we plotted average friend count across the ages. Let's see if you can change the code for this plot to plot the means instead of the medians for each cohort. You'll also want to plot out the grand mean as well, that we talked about in lesson four.



Answer:

To plot the means and study the medians for each of the cohorts, we just need to change our fun.y parameter. This needs to be mean instead. Now, to get the grand mean, we just add a geom line and then set the parameters. We'll have a stat of summary. We'll set fun.y to equal the mean, and then we'll set the line type equal to two. Here, I'm using two so that way, I get a dash line so it stands out in comparison to these others. Plotting the grand mean is a good reminder that much of the data in the sample is about members of recent cohorts. This is the type of more high level observation that you want to make as you explore data.

Programming Quiz: Friending Rate

Since the general pattern continues to hold after conditioning on each of the buckets of year joined, we might increase our confidence that this observation isn't just an artifact of the time users have had to accumulate friends. Let's look at this relationship in another way. We could also look at tenure and friend count as a rate instead. For example, we could see how many friends does a user have for each day since they've started using the service. Let's see if you can create a summary of this rate, that shows how many friends a user has for each day since the user started using Facebook. Subset the data so you only consider users with at least one day of tenure. Once you have that summary answer these two questions. What's the median rate? And, what's the maximum rate?

Answer:

Here we want a summary of the friend rate. So, we can use the with command and subset the data so we only consider users with tenure of at least one day. Now we just want a summary of friend count divided by tenure, which gives us the friends per day since the user's been active. When we run the code, we see that the median rate is about 0.22, and the maximum rate is 417. Now, this is definitely

Copyright © 2014 Udacity, Inc. All Rights Reserved.

an outlier, considering our data, since the third quartile is only about 0.5.

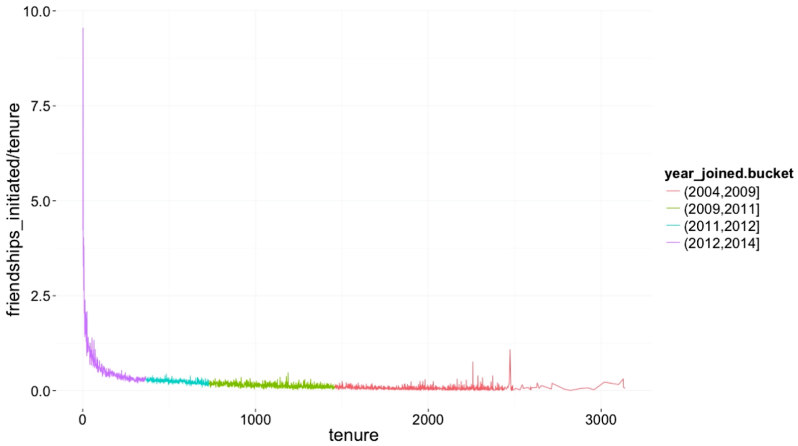
Programming Quiz: Friendships Initiated

We've learned even more about the relationship between age and friend count by making use of the variables' gender and tenure. We've seen that users who have been on the site longer, typically have higher friend counts across ages. Now, this leaves me wondering if friend requests are the same or different across groups. Do new users go on friending sprees? Or do users with more tenure initiate more friendships? Let's explore this with a plot. I want you to create a line graph of friendships initiated per day, versus tenure. You need to make use of the variables age, tenure, friendships initiated, and year_joined.bucket. The color of each part of your one line should correspond to each bucket of year_joined. You'll also need to subset the data to only consider users with at least one day of tenure.

Answer:

To create this plot, we're going to have much of the same code as before. We'll pass tenure to our x variable, and then we'll pass friendships initiated divided by tenure to our y variable, and we'll subset our data frame so that we only consider users who have a tenure of at least one day. We'll color our line by year_joined.bucket and then we'll plot the mean of the y variable across tenure. Taking a closer look, it appears that users with more tenure typically initiate less friendships.

Programming Quiz: Bias Variance Trade off Revisited



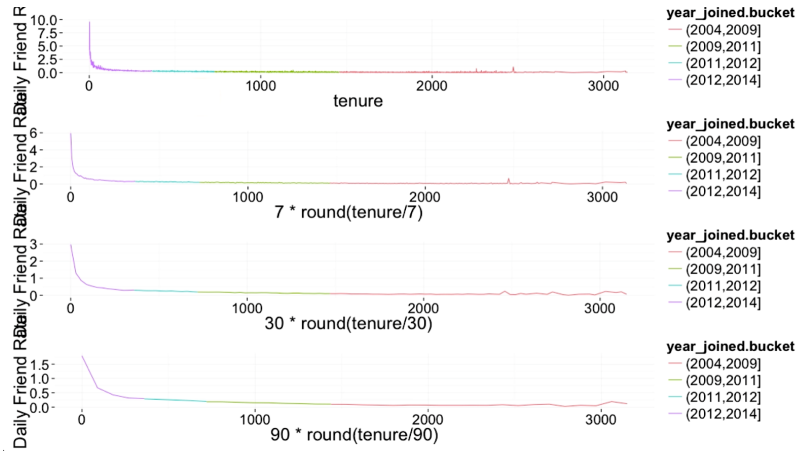
There's a lot of noise in our graph since we are plotting the mean of y for every possible tenure x value. Recall from lesson four that we can adjust this noise by bending our x-axis differently. Let me show you one of those changes in the bend width. Here's our code from before, and instead of using tenure here, I'm going to replace this with a different version or formula so that way I can bend some of the tenures together. Now let's see the difference when I plot this graph. Notice how I have slightly less noise in my plot. We still see some of the same peaks from before, especially here, but it's much

Copyright © 2014 Udacity, Inc. All Rights Reserved.

smoother in general. Here's another one using the number 30 instead of the number seven and here's a graph with very high bias, but much less variance using the number 90. Here I plotted all the graphs so we could compare it to the original one. Notice that as the bin size increases we see less noise on the plot. Our estimates are adjusted since we have more data points for our new values of tenure. In lesson four we introduce smoothers as one tool for an analyst to use in these types of situations. So instead of using gm line here, I'd like you to use gm smooth, to add a smoother, to this plot. Try doing this, in this next programming assignment.

Answer:

To add a smoother to this plot, we need to change geom line to geom smooth. We also need to get rid of this fun.y parameter and the stat parameter. We'll still keep year_joined.bucket assigned a color, so that way we see this segment in our graph. And here I'm using the defaults for geom smooth. So R will automatically choose the appropriate statistical methods based on our data. All of that additional information can be found in the output down here. So here again, in this smooth version of the graph, we still see that the friendships initiated declines as tenure increases.



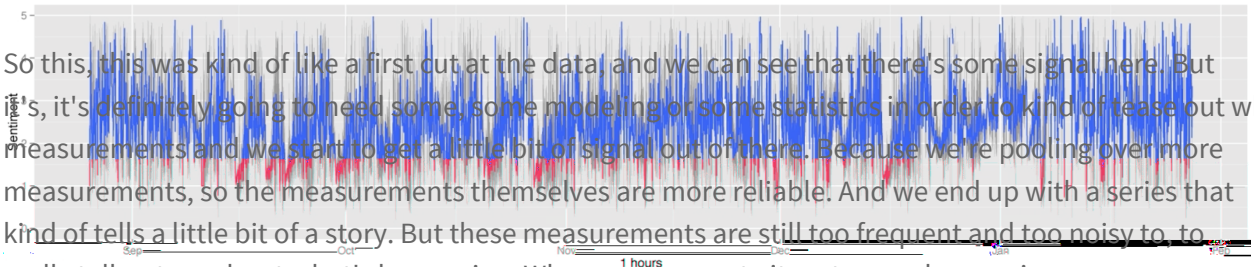
Seans NFL Fan Sentiment Study

You learned about the bias- ‘‘variance tradeoff in this lesson and the last lesson. Now I want you to hear from Sean. Hear about his work at Facebook and pay careful attention to his models and the trade offs in the visualizations he made.

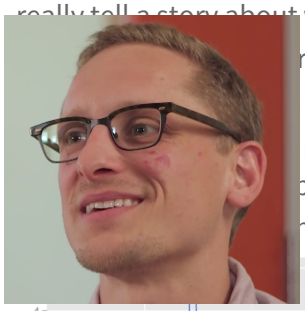
Some recent work I've been doing on measuring fan sentiment for NFL teams over the course of the season that I'm really excited about. it's, it's been a really fun project be I myself am, am an NF, NFL fan. I'm a huge Eagles fan and I go through all of the, emotions tha goes through over the course of the season. You know, the highs of when your team wins and t the lows after a couple losses in a row of feeling kind of hopeless. And so I got the idea of mayb could measure this and kind of tell a story. Not just for my team but for all the other teams in t and come up with some idea, some way of visualizing this, this experience of being a fan. And s counted ratios of positive to negative words at five-minute increments over the course of the w months of the NFL season. And because we're taking a ratio, we end up with some measurements that are extremely

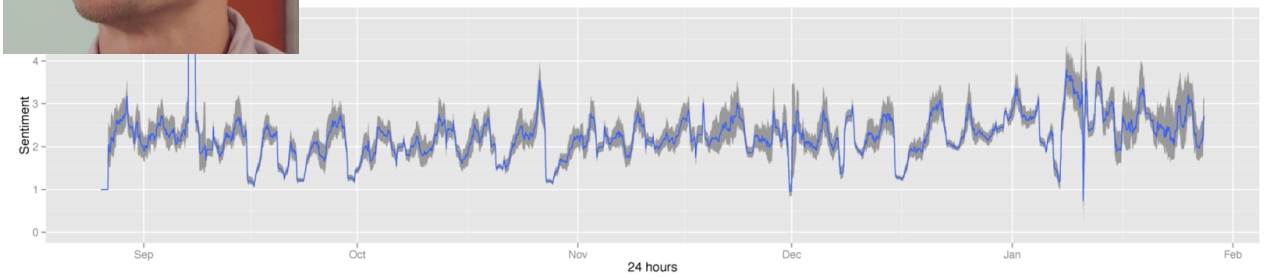
Copyright © 2014 Udacity, Inc. All Rights Reserved.

high. Like, you know, positive to negative word ratios, oh, over a hundred even though the average in somewhere in the two to three range.



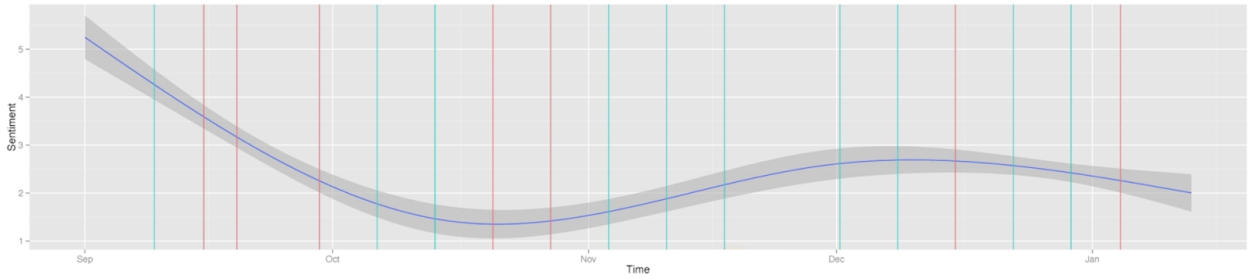
So this, this was kind of like a first cut at the data, and we can see that there's some signal here. But it's, it's definitely going to need some, some modeling or some statistics in order to kind of tease out what's actually ha measurements and we start to get a little bit of signal out of there. Because we're pooling over more measurements, so the measurements themselves are more reliable. And we end up with a series that kind of tells a little bit of a story. But these measurements are still too frequent and too noisy to, to really tell a story about what's happening. When we aggregate it up to one day moving averages, we rns emerge. I guess one of the key features of this dataset was that I knew what of time. Because I, I am an Eagles fan, I experienced the highs and lows of the plot like this immediately and tell you this is not telling the story that I want. pply some modeling to that. We want a model that predicts sentiment as a ne of the things that comes to mind right away is a natural spline.



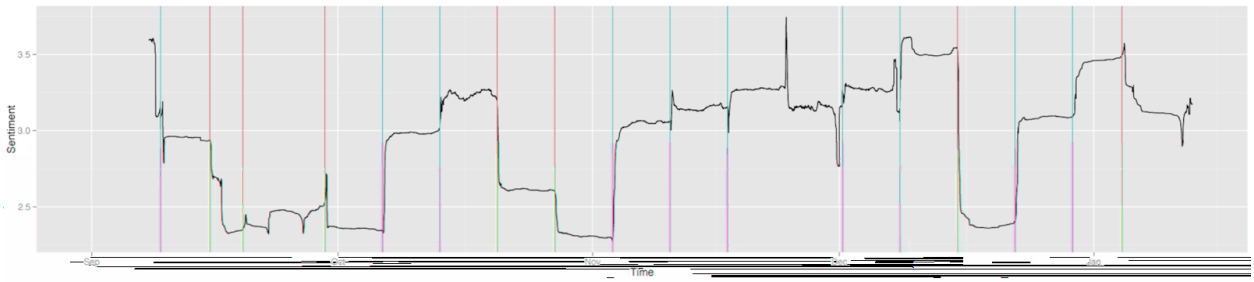


You can see that this actually tells a nice story. These color lines are the dates of wins and loses. It kind of gives you an idea of why the lines are upward sloping or downward sloping. So you can see here, kind of like the exuberance at the beginning of the season as people are really optimistic. And then, you know, three losses in a row and see how the sentiment dips. This tells a nice story, but it doesn't have the feature of, that we'd expect, which is that on game day, things change really abruptly. At the very end of the game, when you know if you've won or you've lost, you're much happier or much sadder than you were at the beginning of the game. So we expect to see really discrete jumps in sentiment that we don't see from a model like this. And this is because this is a bad model of the underlying data generating process.

Copyright © 2014 Udacity, Inc. All Rights Reserved.

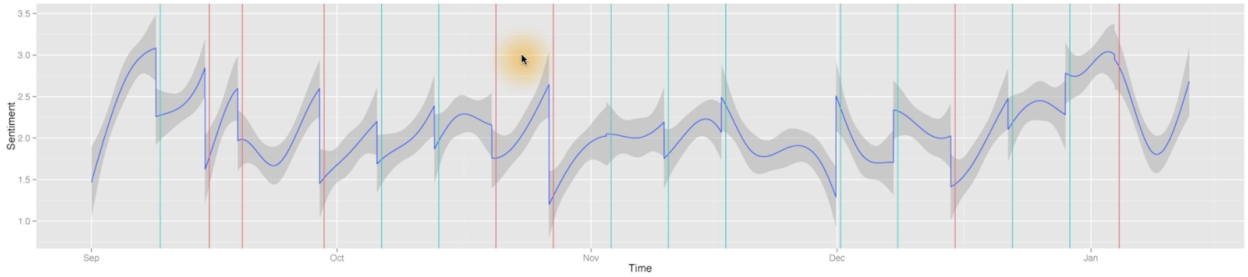


So we need something more flexible. One way to do that is just to use a seven day moving average which is going to allow us to include only kind of the last game sentiment in the moving average. So we're going to pick a moving average, like I showed you before. Let's smooth it out into over a whole seven day period. And when we do that, we get something that actually tells a really nice story about the season. And has all the kind of characteristics that I would expect as a fan in having gone through this. Which is, the kind of, the big bumps up on game days where you win, the big bumps down on game days where you lose. And then, kind of, these plateaus in between, which are these periods of stability when you don't have any information about how your team is doing. We see that a week off around Thanksgiving but then there's a big spike in happiness because people are just happy around Thanksgiving. This big low point right after a loss that could have knocked them out of the playoffs. And then a big kind of ascension to their playoff game which they ended up losing and the subsequent dip. This I think is a really nice depiction of what happened and it took a little bit of averaging to get the story to come out.



When you're looking at all this data, what sort of things come up for you in terms of bias and variance tradeoffs. When you're computing just a simple moving average like this, you're dealing with one of the most, it's just a really flexible statistic. And so you're, not imposing any structure on the data. You're letting the data kind of speak. When I use a moving average here and I plotted standard errors that were kind of rolling along with the data. They were gigantic. The mean sentiment for the season is somewhere in the three range. And the standard errors were over two or three. We can say very precisely what's happening at any given point. But our variance on that estimate is huge. So as we started to add more lags, higher number of lags to the, moving average. We end up with kind of smoother looking plots, that have lower variance but but then are getting progressively more bias.

Copyright © 2014 Udacity, Inc. All Rights Reserved.



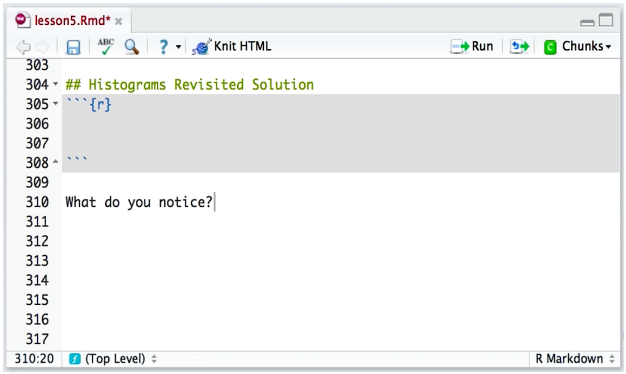
So as we go back further we are including more data and were getting more bias. Because were including data from parts that actually aren't applicable to that exact point. But in exchange for that we get a lower variance plot, one that doesn't move as wildly. When we combine that with splines, we can end up fitting a model, that has kind of the best of both worlds. Which has the smoothing, aspects of the splines, with the discrete jumps, of what happens on game day. And so this is a spline where we add it dummy variables for post game periods. In this model, we end up with kind of all the same thing. We get the big jumps that we expect, so it jumps down on loses, jumps up on game days where they win. And then also kind of the smooth transitions in between. So it's kind of a nice story of taking one style of model, which is a spine, which is just too specific for the data generating process, and maybe not a good fit. And in doing some exploratory data analysis where we see that averaging over seven days tells a really nice story and gives us the discreteness that we want. And then combining those two together into kind of an aggregate of the two types of models. Where We're able to better account for the fact that game days, are, are an important thing.

Introducing the Yogurt Dataset

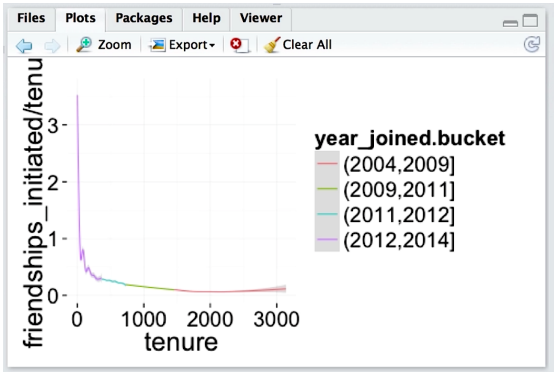
Throughout all of our analyses of the pseudo-Facebook user data set, we've come to learn a lot about our users. From their birthdays, to their friend counts, to their friendships initiated, we've really come to understand their behaviors and how they use the Facebook platform. But, now I think it's time for something completely different. In the next couple of segments, we'll look at another data set, and then we'll return to this Facebook data set to draw some comparisons Here's Dean to introduce this new data set, and why we might look at it in the first place. >> Because of online purchases, credit cards, and loyalty cards, lots of retail purchase data is associated with individuals or households, such that there is a history of purchase data over time. Analysts in industry often mine this panel scanner data, and economists and other behavioral scientists use it to test and develop theories about consumer behavior. We are going to work with a data set describing household purchases, of five flavors of Dannon yogurt in the eight-ounce size. Their price is recorded with each purchase occasion. This yogurt data set has a quite different structure than our pseudo-Facebook data set. The synthetic Facebook data has one row per individual with that row giving their characteristics and counts of behaviors over a single period of time. On the other hand, the yogurt data has many rows per household, one for each purchase occasion. This kind of microdata is often useful for answering different types of questions than we've looked at so far.

Copyright © 2014 Udacity, Inc. All Rights Reserved.

Programming Quiz: Histograms Revisited



I'd like you to get started by looking at the structure and the summary of the Yerger data set on your own computer. You may want to pause the video now to give yourself some time to load in the data set and to download it first from the instructor notes. This Yerger data set contains over variables. The observations are for households that buy Dannon yogurt over time. Now one thing that you might notice is that most of the variables in here are integers. But I want to convert one of the variables to a factor, and that's the ID variable. We'll see how this comes in handy later in the course. But just make sure it says factor right here on your data set as well. If ID doesn't say factor for the type of variable, make sure you run this command in order to change ID to a factor variable. Once you've done all this, I want you to create a histogram of the Yogurt prices. Copy and paste your R code in to the R chunk on the quiz screen that will appear here. And then write a few sentences about what you notice. So that way we can be consistent as we work through the data set together, I want you to be sure to read in the data set into the variable called yo. You can, of course, use other variable names. But our auto grader will only accept answers for programming assignments that have this data frame labeled as yo.



Answer:

This code should be really familiar to you because it's what we saw in lesson three. One of the first plots you created was a histogram, so let's see what this one looks like. We can immediately notice some important discreetness to this distribution. There appear to be prices at which there are many observations, but then no observations in adjacent prices. This makes sense if prices are set in a way that applies to many of the consumers. There are some purchases that involve much lower prices, and if we are interested in price sensitivity, we definitely want to consider what sort of variations is in these prices. Now, I also want you to note that if we chose a different bin width we might obscure this discreetness. Say if I chose a bandwidth equal to ten. In this histogram, we would miss the observation for some of the empty spaces for the adjacent prices. So, it's no surprise that for this very discreet data this histogram is a very biased model.

Copyright © 2014 Udacity, Inc. All Rights Reserved.

Programming Quiz: Number of Purchases

There are other ways we could have noticed this important feature of the data set. But, there are also other ways, that by jumping into a different analysis, we might have missed this. For example, if we just look at a five number summary of the data, we might not notice this so easily. One clue to the discreteness is that the 75th percentile is the same as the maximum. We could also see this discreteness by looking at how many distinct prices there are in the data set. So here it looks like there's about 20 different prices. Tabling the variable we get an idea of the distribution like we saw in the histogram. So now that we know something about the price, let's figure out on a given purchase occasion how many eight ounce yogurts does a household purchase. To answer this we need to combine accounts of the different yogurt flavors into one variable. For example, for this particular household, on one purchase occasion, they bought three different types of yogurt. To figure this out for all the households, we need to make use of a new function. The function is called the transform function. Now I haven't introduced this function yet, but I want you to look up the function and run the examples of the code at the bottom of the documentation to figure out what it does. See if you can create a new variable called all.purchases, which gives the total counts of yogurt for each observation or household purchase. Keep in mind that you need to save this variable to the data frame, so this may or may not be in your syntax.

Answer:

For this programming assignment you needed to create the variable all purchases and add it to our Yo data frame. The transform function takes in a data frame and allows us to add different variables to it by recombining variables that are already within the data frame. So here I'll create the variable All Purchases and then I'll set it equal to the sum of all the yogurt flavors strawberry, blueberry, pina colada , plain, and mixed berry. When I run the code, I can see that my number of variables changed. And, just to be sure, I can print a summary of this variable as well. You might have also used this code. This code is fine, it's just a bit more verbose than we need.

Programming Quiz: Prices Over Time

Now that we have an extra column, our variable in our data frame, we can create a histogram of it. This histogram reveals that most households buy one or two yogurts at a time. To dive deeper into

Copyright © 2014 Udacity, Inc. All Rights Reserved.

our yogurt prices and household behavior. Let's investigate the price over time in more detail. This is where our Facebook data was deficient. In this data set, we can examine changes in prices because we have data on the same households over time. So, now I want you to use your knowledge of scatter plots and over-plotting to create an appropriate visualization. Your visualization should be a scatter plot of price versus time. Now, there's no right answer here, so just create a plot that's sensible and that allows you to see the data. You can assess your code and your plot with what we did in the solution video.

Answer:

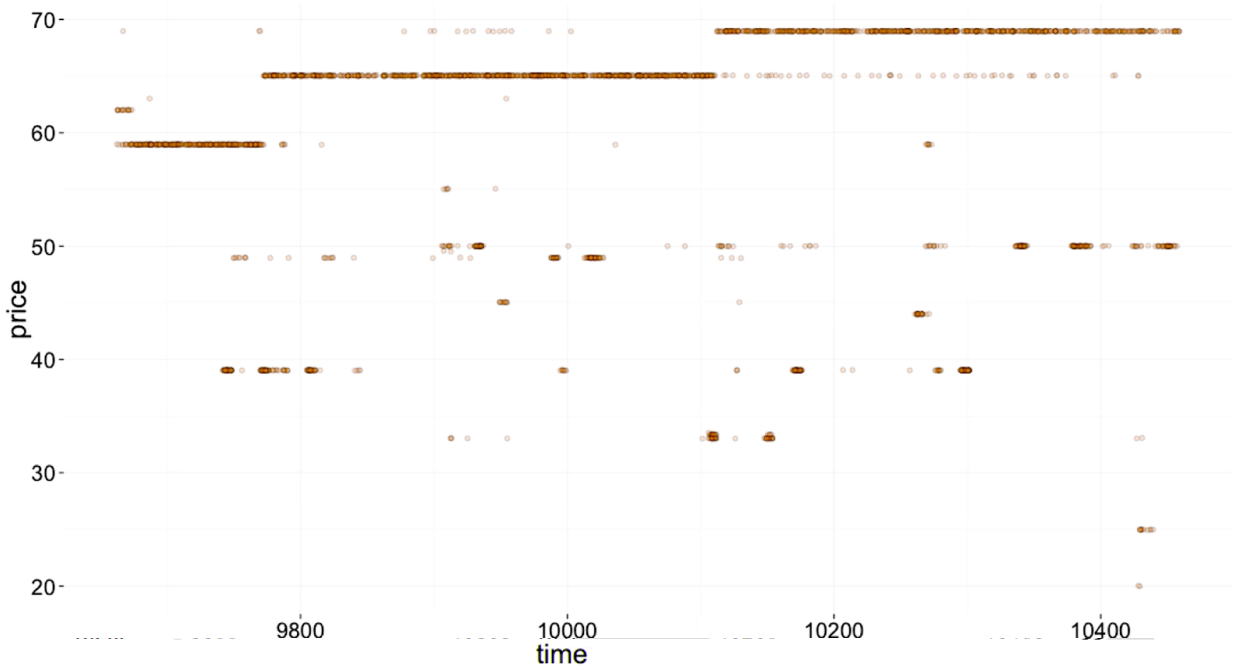
For this solution, I'm showing you the ggplot code to create the scatter plot of price versus time. Now, I've added geom jitter here. So, that way, my points will show up a little bit transparent. And I'll color them with an orange hue. Looking at the plot, we can that the mode or the most common prices, seem to be increasing over time. We also see some lower price points scattered about the graph. These may be due to sales or, perhaps, buyers using coupons that bring down the price of yogurt.

Sampling Observations

Now I know our work up to this point with this yogurt data set has been review and it should be, but lets hear from Dean about how we might proceed differently with this type of a data set.

When familiarizing yourself with a new data set that contains multiple observations of the same units, it's often useful to work with a sample of those units so that it's easy to display the raw data for that sample. In the case of the yogurt data set, we might want to look at a small sample of households in more detail so that we know what kind of within and between household variation we are working with.

Copyright © 2014 Udacity, Inc. All Rights Reserved.

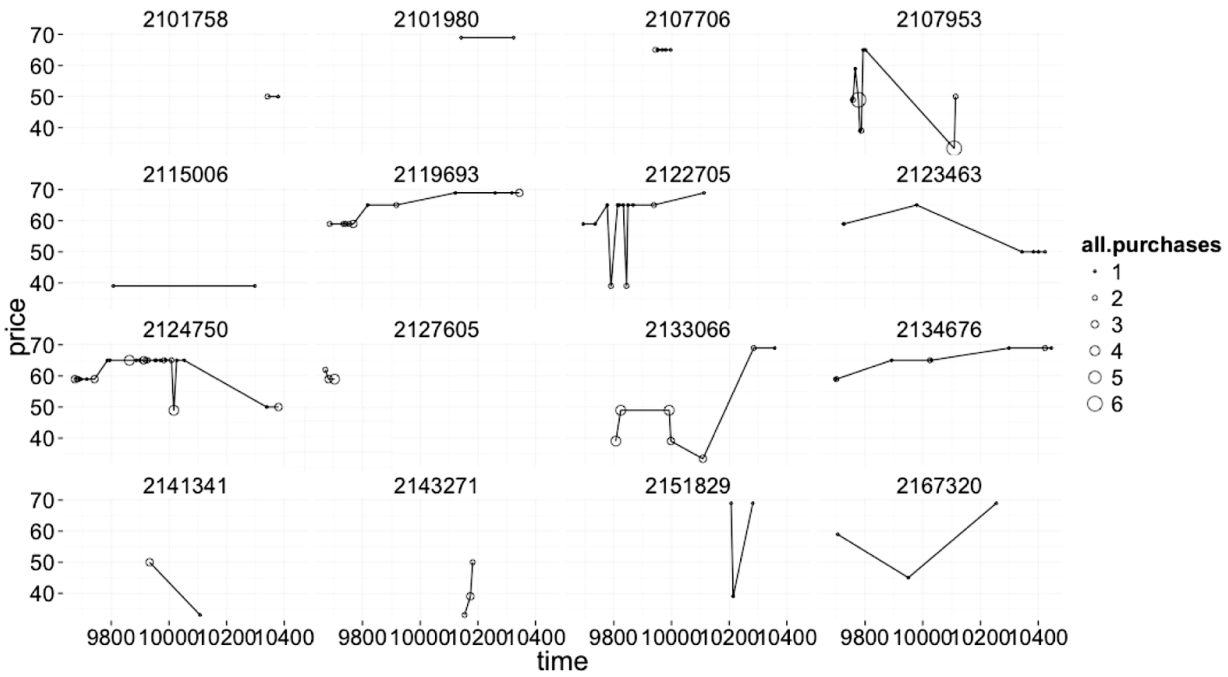


This analysis of a sub-sample might come before trying to use within household variation as part of a model. For example, this data set was originally used to model consumer preferences for variety. But, before doing that, we'd want to look at how often we observe households buying yogurt, how often they buy multiple items, and what prices they're buying yogurt at. One way to do this is to look at some sub-sample in more detail. Let's pick 16 households at random and take a closer look.

Programming Quiz: Looking at Samples of Households

Thanks again, Dean. Let's cut up the exploration Dean just described. First, we should use the set seed function to make this reproducible. Running this one line of code will also allow you to see the same households in my explorations, if you use the same seed number. Now, let's sample 16 of the households from our yogurt data set, from yogurt ID. Notice that I'm sampling from the levels because those are all of the different households that I have. I'll run the code for the sample ID's. And then, I'll print out the sample ID's. There's the 16 of them. Now we can plot each purchase occasion for each of the households that we sampled. We have the time of the purchase, the price per item of the yogurt, and the number of items. Here, I'm using the size parameter to add more detail to my plot. I'm passing at the all purchases variable, so that way I can consider the number of items in terms of size of the point on the plot. So, before I run this code, I want you to take a minute to think about what type of plot this code will produce. You maybe want to pause for a moment and give yourself some time to think. Now you might also notice this new syntax here. This percent in percent sign. This just tells us to loop over the IDs, so that when we go through and create a panel plot, for each ID of our households. And here's the plot. Notice how we get panels for each of the households that we had in our sample.

Copyright © 2014 Udacity, Inc. All Rights Reserved.



From these plots, we can see the variation and how often each household buys yogurt. Here, a lot and here, not to much. And it seems that some household purchases more quantities than others with these larger circles indicating not here. For most of the households, the price of yogurt holds steady, or tends to increase over time. Now, there are, of course, some exceptions, like in this household and in this household, and even here, we might think that the household is using coupons to drive the price down. Now, we don't have the coupon data to associate with this buying data, but you could see how that information could be paired to this data to better understand the consumer behavior. So now, let's have you do something similar. Your task is to create a similar plot and provide your own insights for another set of your seed number and the title and in the post, you can include an image of your plot and any observations that you make. You may want to try this for a few households to get a better feel for the data and the plot. There won't be any solution video here, so we'd love to hear from you all and see you all discuss what's going on in the discussions.

Answer:

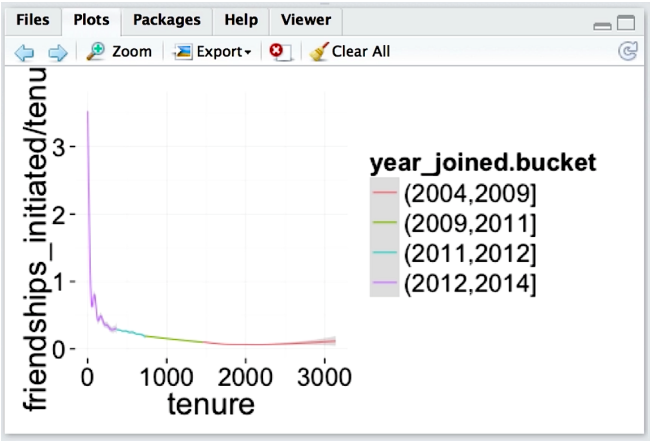
The general idea is that if we have observations over time, we can facet by the primary unit, case, or individual in the data set. For our yogurt data it was the households we were faceting over. This faceted time series plot is something we can't generate with our pseudo Facebook data set. Since we don't have data on our sample of users over time. Let's get back to that plot to see that limitation. The Facebook data isn't great for examining the process of friending over time. The data set is just a cross section, it's just one snapshot at a fixed point that tells us the characteristics of individuals. Not the individuals over, say, a year. But if we had a dataset like the yogurt one, we would be able to track friendships initiated over time and compare that with tenure. This would give us better evidence to explain the difference or the drop in friendships initiated over time as tenure increases.

Copyright © 2014 Udacity, Inc. All Rights Reserved.

Many Variables

Before we wrap up our work with the pseudo-Facebook data, let's hear from Dean again about another technique that you can use in your work as a data analyst.

Much of the analysis we've done so far focused on some pre-chosen variable, relationship or question of interest. We then used EDA to let those chosen variables speak and surprise us. Most recently, when analyzing the relationship between two variables we look to incorporate more variables in the analysis to improve it. For example, by seeing whether a particular relationship is consistent across values of those other variables. In choosing a third or fourth variable to plot we relied on our domain knowledge. But often, we might want visualizations or summaries to help us identify such auxiliary variables. In some analyses, we may plan to make use of a large number of variables. Perhaps, we are planning on predicting one variable with ten, 20, or hundreds of others. Or maybe we want to summarize a large set of variables into a smaller set of dimensions. Or perhaps, we're looking for interesting relationships among a large set of variables. In such cases, we can help speed up our exploratory data analysis by producing many plots or comparisons at once. This could be one way to let the data set as a whole speak in part by drawing our attention to variables we didn't have a preexisting interest in.

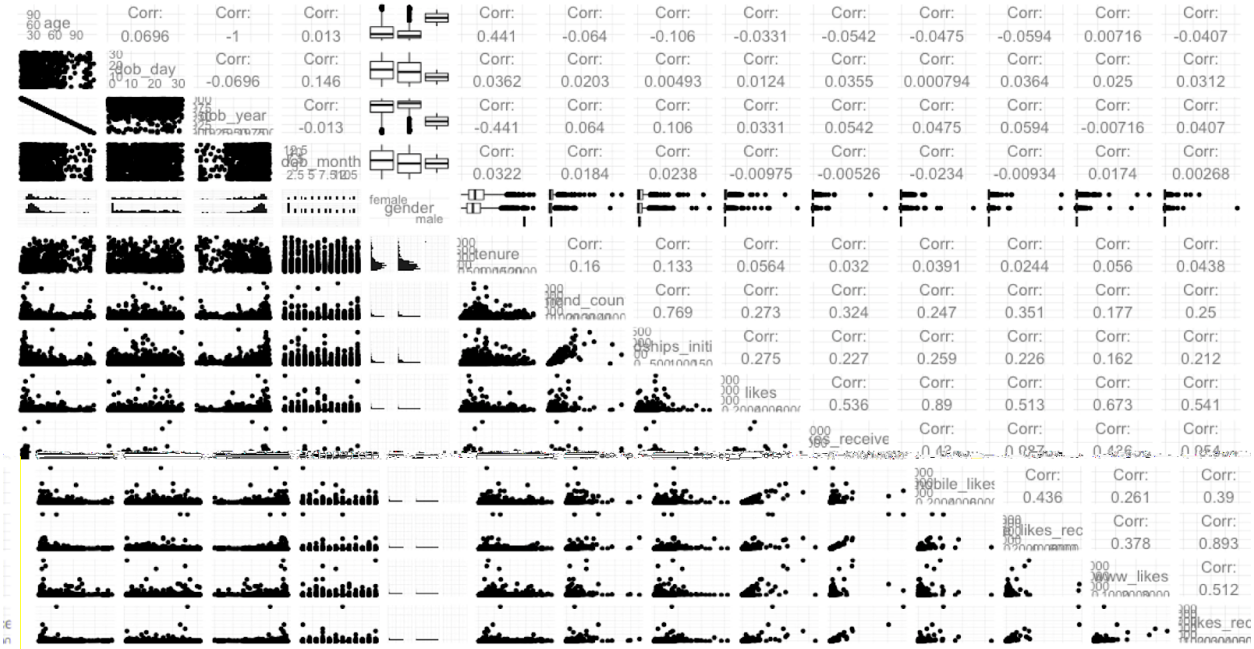


Programming Quiz: Scatterplot Matrices

As Dean described, we should let the data speak to determine variables of interest. There's a tool that we can use to create a number of scatter plots automatically. It's called a scatter plot matrix. In a scatter plot matrix. There's a grid of scatter plots between every pair of variables. As we've seen, scatter plots are great, but not necessarily suited for all types of variables. For example, categorical ones. So there are other types of visualizations that can be created instead of scatter plots. Like box plots or histograms when the variables are categorical. Let's produce the scatter plot matrix for our pseudo Facebook data set. We're going to use the GGally package to do so. So make sure you've installed it and then go ahead and load it using the library command. Now, I'm also going to set the theme here too. Now, there's two other things that we want to do. First we want to set the seed so we get reproducible results. Now, you might be wondering why we set the seed in the first place. And it's because we're going to sample from our data set. Our data set contains all these variables and I

Copyright © 2014 Udacity, Inc. All Rights Reserved.

actually don't want all the variables. I don't want user ID, year joined, or year joined.bucket. So what I can do is subset my data frame and then sample from that subset. If I check out the variables in my subset data frame these are the ones of interest.



Now I didn't use year joined or year joined.bucket, because this one's a categorical variable and really these were derived from tenure. Now I'm ready to use the GG pairs function inside of GGally to create this scatter plot matrix. Now, I've already run this piece of code and I do want to warn you that it takes a long time for this to generate. It might even take over an hour. Feel free to run the command and if its taking a long time for your plot to generate just come back to your computer at another time. We've also included a pdf of the scatter plot in the instructor notes so you can check that out as well. Here's our scatter plot matrix, and notice in the upper part of the matrix we can see the correlation coefficients for the pairs of variables. For age and date of birth year, the correlation is actually negative one. And we can see that on the scatter plot. Sometimes we may want to produce these types of matrices so that way we can produce one number summaries of the different relationships of our variables. This is just like the correlation work that we did in lesson four. So, I've described the plots above the diagonal for the scatter plot matrix, but what do you notice in the lower left of the scatter plot matrix? Write a few sentences about what you see in this next quiz. And pay careful attention to the variable of gender. What types of plots do you think these are?

Answer:

The GG pairs function uses a different plot type for different types of combinations of variables. Hence, we have histograms here and we have scatter plots here. Many of these plots aren't quite as nice as they would be if we fine-tuned them for the particular variables. For example, for all the counts of likes, we might want to work on a logarithmic scale. But, ggpairs doesn't do this for us. At the very

Copyright © 2014 Udacity, Inc. All Rights Reserved.

least, a scatter plot matrix can be a useful starting point in many analyses.

Even More Variables

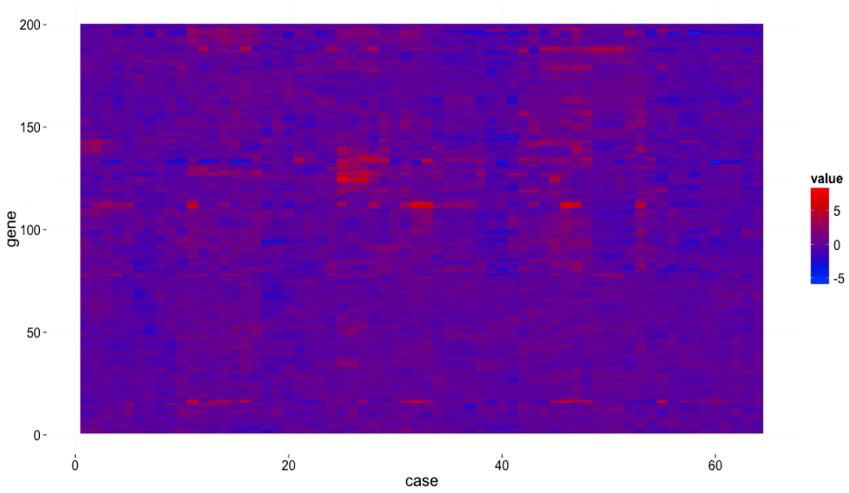
A matrix such as this one will be extremely helpful when we have even more variables than those in the pseudo-Facebook data set. Examples arise in many areas, but one that has attracted the attention of statisticians is genomic data. In these data sets, they're often thousands of genetic measurements for each of a small number of samples. In some cases, some of these samples have a disease, and so we'd like to identify genes that are associated with the disease. In the instructor notes, you'll find a data set of gene expression in tumors. The data contains the expression of 6,830 genes, compared with a larger baseline reference sample. Now, this is a ton of data. So let's go ahead and read in the data set, and then I'll change the color names of the data set to be the numbers from one to 64. Now, I'm just doing this so that way the plot that I create is going to be a little bit nicer with the labeling on the x axis.

Heat Maps

The last plot that we'll make for this course is called a Heat Map. For our data set we want to display each combination of gene and sample case, the difference in gene expression and the sample from the base line. We want to display combinations where a gene is overexpressed in red. in combinations where it is under expressed in blue. Here's the code to make that Heat Map. First, we'll run all of this in order to melt our data to a long format. And then we just run our ggplot code using the geom, geom tile. Now, this last line is going to give us a scale gradient. And we're going to use the colors from blue to red. So, let's see what the output looks like. And, there's our Heat Map. Even with such a dense display, we aren't looking at all the data. In particular, we're just showing the first 200 genes. That's 200 genes of over Genomic data sets of these kind, sometimes called micro data are only getting larger, and more complex. What's most interesting, is that other data sets also look like this. For example, internet companies run lots of randomized experiments. Where in the simplest versions, users are randomly assigned to a treatment like a new version of a website or some sort of new feature or

Copyright © 2014 Udacity, Inc. All Rights Reserved.

product or a control condition. Then the difference in outcome between the treatment and control can be computed for a number of metrics of interest. In many situations, there might have been hundreds or thousands of experiments and hundreds of metrics. This data looks very similar to the genomic data in some ways. And this is why the useful maxim plot all the data might not always apply to a data set as it did to most of this course.



Reflection Quiz: Analyzing Three or More Variables

Congratulations. You've made it a long way and you've reached the end of lesson five. I want to end this lesson with another reflection. Take a few moments and write down some of the things you've learned in this lesson in the text box that will appear. Once you've done that, submit your answer and then hear from Tim and Moira about what they thought was important.

Analyzing Three or More Variables

Congratulations on finishing lesson five. Here, we took many of the basic techniques you learned in previous lessons and extended them to look at patterns across many variables at once. >> We started with simple extensions to the scatter plot, and plots of conditional summaries that you worked with in lesson four, such as adding summaries for multiple groups. Then, we tried some techniques for examining a large number of variables at once, such as scatter-plot matrices and heat maps. >> We also learned how to reshape data, moving from broad data with one row per case, to aggregate data with one row per combination of variables, and we moved back and forth between long and wide formats for our data. >> Next up in lesson six, we'll hear from our colleague Solomon. He's going to do an in-depth analysis of the diamond data set, so you can see how an expert does it. Lesson six also covers a larger part of the data analysis process, highlighting the role of EDA. Solomon did a lot of reading about the diamond industry. He wrote codes to scrape and format a new data set, and he used the results of his exploratory data analysis to guide, interpret, and evaluate a statistical model of diamond prices.

Copyright © 2014 Udacity, Inc. All Rights Reserved.