Homework 2

Due by 9PM, September 20th

Please submit R code, explanations, and / or plots for any section marked with the **[To submit]** heading to jakeporway+itp@gmail.com.

READINGS

Background Reading for This Homework

- Go back and read through the notes in Lecture 1 and Lecture 2 (http://jakeporway.com/teaching/notes and http://jakeporway.com/teaching/code). Sadly we don't get to everything in class, but you're expected to know everything there so start off on the right foot by going through those notes.
- Speaking of, can you believe I never mentioned in the notes what table() does exactly? Hopefully you've heard me say it, read the online help, or seen it in the notes, but it's super useful – it adds up the number of times each variable occurs in a vector. We're going to use it a lot any time we want to get a count of the number of times something happens.
- I know this seems like a long homework, but it's really not bad. Most of it is expository text so don't lose hope!

Reading for Next (or Next Next) Class

I don't know how much of this we'll get to next class, but these are some really important topics that we'll be touching on soon, if not next week (in descending order of importance):

- Summarizing data in R (part of our exploratory data analysis toolkit): Review pp 1-16 in http://cran.r-project.org/doc/contrib/Verzani-SimpleR.pdf (the printed page numbers, not the PDF page numbers).
- More summarizing data in R: pp 23-27 (also the printed page numbers) in http://cran.r-project.org/doc/contrib/Owen-TheRGuide.pdf.
- Intro to regular expressions: http://www.proftpd.org/docs/howto/Regex.html
- JSON: http://en.wikipedia.org/wiki/JSON

Part One – Stop and Frisk Round 2

Last class we raced through the ins and outs of vectors, data frames, and subsetting based on logical constructions. Though it may have been a bit frenzied, we now have the power to *treat R like a database*. Whoa, that's pretty powerful. Why don't we dig a little deeper into our Stop and Frisk data using our advanced skills to answer some real questions.

Let's use the data we saw in class last time, which you can get from http://www.jakeporway.com/teaching/data/snf_2.csv

Policing the Police:

One of the biggest and most charged questions around Stop and Frisk is whether the police are using racial discrimination or not. We haven't learned the kind of statistical tests that would let us really *prove* (or disprove) this yet, but let's explore that question by starting with a simpler task: analyzing how the police treat different races.

The data we have consists solely of "stops", i.e. a case where a police officer approached someone. Not all of them end up in frisks, searches, or arrests. According to police practice, an officer should only frisk someone when they feel that the person is a threat to the officer. Let's see how threatened NYPD is:

[To submit]: Write code to return the percentage of people who were frisked for each race. In other words, count up the number of people who were frisked for a given race divided by the number of people of that race stopped. Which race leads to the highest percentage of frisks? Which one the lowest?
NOTE 1: You can look up what each race code means at
http://www.jakeporway.com/teaching/resources/SNF_codes.pdf
NOTE 2: The enterprising amongst you might use a loop to do this, but since we haven't technically learned how to do those yet, I'd just type each case in by hand.

To even stop someone, the police need to have some kind of suspicion that the person is doing something illegal. We saw last time that the crime.suspected variable holds information on all the crimes people are suspected of committing (though we also saw that there were A LOT of different crimes). Let's see which crimes each race is most often stopped for.

Before we do that though, let's dig a little deeper into this messy crime.suspected variable. Last time we saw there were 1346 unique crimes! That's crazy. Let's start out by looking at what the most common crimes are.

[To submit] Plot the number of times each crime occurs in descending order (we've learned a couple of ways to do this, though using sort(), table() and that new type= parameter to plot() is your best bet). What does this distribution of crimes look like? In other words, are there an equal number of every kind of crime or are there a few that dominate?

[To submit] Well I'm kind of answering that question for you here – let's take the top 30 suspected crimes and look at those. If we were to just look at stops where the crime.suspected was one of the top 30 crimes, what percentage of the stops would that cover? Do you think that's enough?

One last thing we should do before we see what crimes each race is being stopped for. Did you notice something about the top 30 crimes suspected? Yeah lots of them have similar names! If you didn't see this in HW1, notice that the top two crimes are "FEL" and "FELONY". Hmm, I wonder if those are the same? Let's clean up our crime.suspected variable a little bit by replacing all crimes with just their first 3 letters. That should help consolidate some of our crimes. But, wait, how do we do that?

Learning in the Homework

There's a great function called "substr()". As you might imagine, it takes "substrings" of strings. Like almost all R functions it's vectorized, so you can just pass it a vector of strings, the position of the first letter you want, the position of the last letter you want, and you're good to go. So, for example, to pull the word "awesome" out of this string, you'd do:

```
X <- "R is awesome beyond belief"
substr(X, 6, 12)
```

[To submit] Write code to create a variable called "crime.abbv" that consists of just the first three letters of crime.suspected and show the code to add it to our main data frame. Now what percentage of the stops do the top 30 crime.abbvs account for?

OK, we're *finally* ready to see what the top crimes are for each race.

[To submit] Write code to show the top 3 crimes each race is suspected of (rev(), sort(), and table() are your friends here again, but you'll have to subset the data by race first). Huh. If you do this right, *almost* all the top 3's should be the same, but a few are different. What are these differences?

Part Two – A Picture's Worth 1000 Words

This one's a quickie: We saw in class that we could look at the number of stops happening each day, using our day variable and table(). It was pretty simple to get those counts, plot them, and even color our plot by day of week. Arguably a more important analysis is what time of day is most common for Stop and Frisks. Let's find out!

[To submit]

1. Let's create an "hour" variable that tells us what hour of the day each stop happened during and add it to our dataset. How do we do this? Well we've got a great column of "time" variables that always has the hour in the same place. Use the substr() function we learned about above to strip out the hour, then use as.numeric() from lecture 2 to convert it to a number.
2. Create a line plot (i.e. a plot with type="l") of the stops by hour. Which hour of the day has the most stops? Which hour has the fewest?
   NOTE: I almost forgot – you'll probably use table() to get the stops counted by hour. Table() has its own plotting conventions, so you should use as.vector(table(…)) to convert the results to a vector of counts (see lecture 2). That way we can play with the plotting parameters ourselves.
3. Create the same plot but with points instead of lines. Use a different plotting symbol than the default and color the max point and min points different colors.

Part Three – No More 64oz Containers

[THIS ONE'S A BONUS – YOU DON'T HAVE TO DO IT BUT I WROTE IT AND I LIKED IT SO HERE IT IS. DIVE IN IF YOU LIKE, OTHERWISE WE'LL GO OVER IT IN CLASS]

People are quick to jump on racial and gender discrimination by the NYPD, but one form of discrimination that's rarely brought up is body discrimination. Let's see if heavier people are more or less likely to get arrested (NOTE: We're not at the stage where we're actually proving this or generalizing from this dataset, we just want to see if more or fewer heavier people were arrested than lighter people in our dataset).

The Stop and Frisk dataset actually contains a "build" variable that specifies the suspect's build, but let's get more scientific than that. Because we have a "height" and "weight" variable, let's calculate the BMI for each suspect.

[To submit]

1. We've seen that many of the variables in our dataset have missing values that are either "0" or "999", and height and weight are no exception. First create a subset of the data only consisting of "good" weights and heights. For our purposes, let's create a subset where the weights are all between 90 and 400, and the heights are greater than 40.
2. Add a BMI variable to our dataset, where BMI is computed as (weight)*703/(height*height).
3. The US Government defines people with BMIs 30 or higher to be obese. What percentage of people with BMI's greater than or equal to 30 who were stopped were ultimately arrested? What percentage of people with BMI's less than 30 who were stopped were ultimately arrested? What do you think of this result?