Week 3 Statistical computing

MS 276

September, 2017

okcupiddata

Today we're going to look at cleaned profile data from OkCupid Profile Data for Introductory Statistics and Data Science (Journal of Statistics Education, 2015)

According to the packages' Github site, the data contain

```
OkCupid users who were living within 25 miles
of San Francisco, had active profiles on June 26, 2012, were online in the
previous year, and had at least one picture in their profile.
```

Let's take a look:

```
library(okcupiddata)
names(profiles)
```

```
##  [1] "age"         "body_type"   "diet"        "drinks"      "drugs"
##  [6] "education"   "ethnicity"   "height"      "income"      "job"
## [11] "last_online" "location"    "offspring"   "orientation" "pets"
## [16] "religion"    "sex"         "sign"        "smokes"      "speaks"
## [21] "status"      "essay0"
```

With data such as these, there are so many interesting questions we can consider! However, several will involve some data manipulation skills.

Data Transformations

We're going to identify five important `dplyr` functions that are appropriate for most data manipulation issues.

Filter

```
profiles %>% filter(drugs == "never")
#
profiles %>% filter(!drugs == "never")
#
profiles %>% filter(drugs == "never", sex == "m")
#
profiles %>% filter(drugs == "never" | sex == "m")
#
profiles %>% filter(drugs == "never", height > 70, sex == "m")
#
profiles %>% filter(drugs %in% c("never", "sometimes"))
#
profiles %>% filter(drugs = "never")
#
dftm <- profiles %>% filter(drugs == "never", height > 70, sex == "m")
```

Arrange

```
profiles %>% arrange(age)
#
profiles %>% arrange(-age)
#
profiles %>% arrange(age, height)
```

Select

```
profiles %>% select(age, body_type, diet, drinks)
#
profiles %>% select(age:drinks)
#
profiles %>% select(-age, -diet)
```

Mutate

```
profiles %>% mutate(drug.free = (drugs == "never"))
#
profiles %>% mutate(feet = height/12, older = age > 40)
#
profiles %>% mutate(feet = floor(height/12), inches = height %% 12)
#
profiles %>% mutate(missing.income = is.na(income))
#
profiles %>% mutate(exists.income = !is.na(income))
```

Summaries and grouping

```
profiles %>%
  summarise(ave.age = mean(age), sd.age = sd(age), med.age = median(age),
            min.age = min(age), max.age = max(age), n.subjects = n())
```

```
##    ave.age   sd.age med.age min.age max.age n.subjects
## 1 32.34029 9.452779      30      18     110      59946
```

```
profiles %>%
  group_by(sex) %>%
  summarise(ave.age= mean(age))
```

```
## # A tibble: 2 x 2
##     sex  ave.age
##   <chr>    <dbl>
## 1     f 32.81822
## 2     m 32.01859
```

```
profiles %>%
  group_by(sex, drugs) %>%
  summarise(ave.height = mean(age), n.subjects = n())
```

```
## # A tibble: 8 x 4
## # Groups:   sex [?]
##     sex     drugs ave.height n.subjects
##   <chr>     <chr>      <dbl>      <int>
## 1     f     never   33.99602      15829
## 2     f     often   24.78125        128
## 3     f sometimes   29.28460       2695
## 4     f      <NA>   31.33760       5465
## 5     m     never   32.61626      21895
## 6     m     often   26.57801        282
## 7     m sometimes   30.17371       5037
## 8     m      <NA>   31.75636       8615
```

```
profiles %>%
  group_by(drugs, sex) %>%
  summarise(ave.height = mean(age), n.subjects = n())
```

```
## # A tibble: 8 x 4
## # Groups:   drugs [?]
##       drugs   sex ave.height n.subjects
##       <chr> <chr>      <dbl>      <int>
## 1     never     f   33.99602      15829
## 2     never     m   32.61626      21895
## 3     often     f   24.78125        128
## 4     often     m   26.57801        282
## 5 sometimes     f   29.28460       2695
## 6 sometimes     m   30.17371       5037
## 7      <NA>     f   31.33760       5465
```

```
## 8      <NA>     m   31.75636       8615
```

```
profiles %>%
  filter(orientation == "straight", !is.na(drugs)) %>%
  group_by(drugs, sex) %>%
  summarise(ave.height = mean(age), n.subjects = n()) %>%
  arrange(ave.height)
```

```
## # A tibble: 6 x 4
## # Groups:   drugs [3]
##       drugs   sex ave.height n.subjects
##       <chr> <chr>      <dbl>      <int>
## 1     often     f   24.39189         74
## 2     often     m   26.33333        249
## 3 sometimes     f   29.94034       1760
## 4 sometimes     m   30.08675       4369
## 5     never     m   32.57007      18859
## 6     never     f   34.25822      14236
```

Putting it all together

```
library(nycflights13)
flights.summary <- flights %>%
  group_by(dest) %>%
  summarise(ave.delay = mean(arr_delay, na.rm = TRUE), n.flights = n()) %>%
  filter(n.flights > 100) %>%
  arrange(-ave.delay) %>%
  slice(1:10)

ggplot(flights.summary, aes(reorder(dest, ave.delay), ave.delay)) + geom_bar(stat = "identity")
  labs(title = "Top 10 worst airports to fly into from NYC in 2013",
       subtitle =  "Among destinations with at least 100 flights",
          x = "Destination code", y = "Average arrival delay")
```