Lab Worksheet 3

Problem 1: *The data set `AirPassengers` built into R lists total numbers of international airline passengers, 1949 to 1960.*

```
AirPassengers
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 1949 112 118 132 129 121 135 148 148 136 119 104 118
## 1950 115 126 141 135 125 149 170 170 158 133 114 140
## 1951 145 150 178 163 172 178 199 199 184 162 146 166
## 1952 171 180 193 181 183 218 230 242 209 191 172 194
## 1953 196 196 236 235 229 243 264 272 237 211 180 201
## 1954 204 188 235 227 234 264 302 293 259 229 203 229
## 1955 242 233 267 269 270 315 364 347 312 274 237 278
## 1956 284 277 317 313 318 374 413 405 355 306 271 306
## 1957 315 301 356 348 355 422 465 467 404 347 305 336
## 1958 340 318 362 348 363 435 491 505 404 359 310 337
## 1959 360 342 406 396 420 472 548 559 463 407 362 405
## 1960 417 391 419 461 472 535 622 606 508 461 390 432
```

Explain the variables present in this dataset. Using the variables in this dataset and the formal definition of tidy data that we learned in lecture, is this data set tidy? Explain why or why not.

The dataset contains the variables for number of passengers, years, and months. The dataset is not tidy. There should be one column for year, one for month, and one for the number of passengers. Instead, the data are arranged such that years vary along the rows and months along the columns.

Problem 2: *The function `data()` lists all datasets that are available in R by default. Look through the list and identify a dataset that is tidy. Explain the variables present in this dataset and why the dataset is tidy.*

I pick the dataset `OrchardSprays` :

```
head(OrchardSprays)
```

```
##   decrease rowpos colpos treatment
## 1       57      1      1         D
## 2       95      2      1         E
## 3        8      3      1         B
## 4       69      4      1         H
## 5       92      5      1         G
## 6       90      6      1         F
```

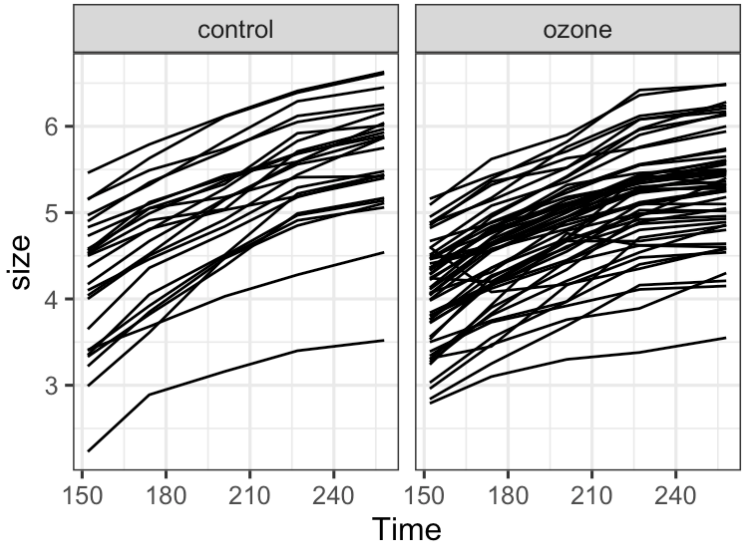
This dataset contains variables for row of the design (`rowpos`), column of the design (`colpos`), treatment level (`treatment`), and response (`decrease`). It contains one row per observation, one column of measured values (`decrease`), and three columns describing the conditions under which that value was measured (`rowpos` , `colpos` , `treatment`).

Problem 3: *In an in-class exercise, we made the following plot of the Sitka dataset:*

```
# download the sitka data set:
sitka <- read.csv("http://wilkelab.org/classes/SDS348/data_sets/sitka.csv")
head(sitka)
```

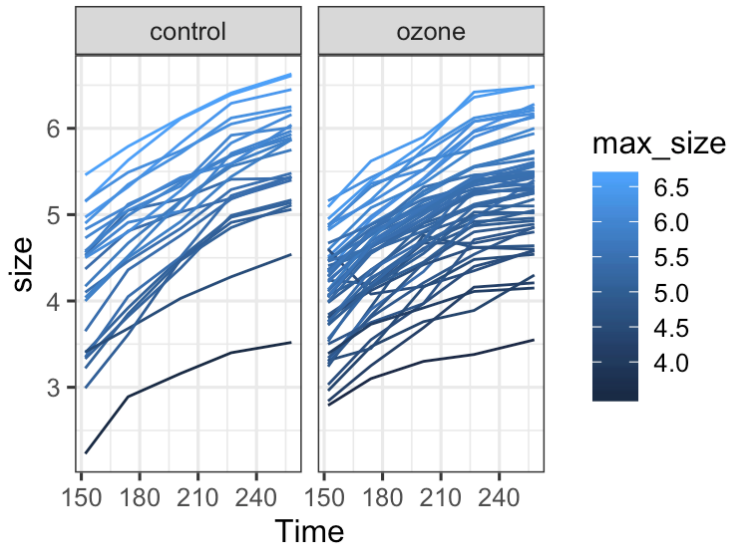
```
##   size Time tree treat
## 1 4.51  152    1 ozone
## 2 4.98  174    1 ozone
## 3 5.41  201    1 ozone
## 4 5.90  227    1 ozone
## 5 6.15  258    1 ozone
## 6 4.24  152    2 ozone
```

```
ggplot(sitka, aes(x = Time, y = size, group = tree)) + geom_line() + facet_wrap
(~treat)
```



Now modify the plot so that the line for each tree is colored according to the maximum size of the tree.

```
sitka_grouped <- group_by(sitka, tree)
sitka_new <- mutate(sitka_grouped, max_size = max(size))
ggplot(sitka_new, aes(x = Time, y = size, group = tree, color = max_size)) + ge
om_line() + facet_wrap(~treat)
```



If that was easy…

Problem 4: *The package nycflights13 contains information about all flights departing from one of the NY City airports in 2013. In particular, the data table* `flights` *lists on-time departure and arrival information for 336,776 individual flights:*

```
library(nycflights13)
flights
```

```
## # A tibble: 336,776 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      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      544            545        -1     1004
##  5  2013     1     1      554            600        -6      812
##  6  2013     1     1      554            558        -4      740
##  7  2013     1     1      555            600        -5      913
##  8  2013     1     1      557            600        -3      709
##  9  2013     1     1      557            600        -3      838
## 10  2013     1     1      558            600        -2      753
## # ... with 336,766 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>
```

We would like to collect some information about arrival delays of United Airlines (UA) flights. Do the following: pick all UA departures with non-zero arrival delay and calculate the mean arrival delay for each

of the corresponding flight numbers. Which flight had the longest mean arrival delay and how long was that delay?

```
flights_filtered <- filter(flights, carrier == "UA" & arr_delay != 0)
flights_grouped <- group_by(flights_filtered, flight)
flights_summary <- summarize(flights_grouped, mean_delay = mean(arr_delay))
filter(flights_summary, mean_delay == max(mean_delay))
```

```
## # A tibble: 1 x 2
##   flight mean_delay
##    <int>      <dbl>
## 1   1510        283
```

Flight 1510 had the longest delay, with an average arrival delay of 283 minutes.