Problem Set #05 Solutions Statistics 506

Problem Set #05

Problem 1 - Plotting

```
library(tidyverse)
library(ggplot2)
nnmaps <- read_csv("data/chicago-nmmaps.csv")
```

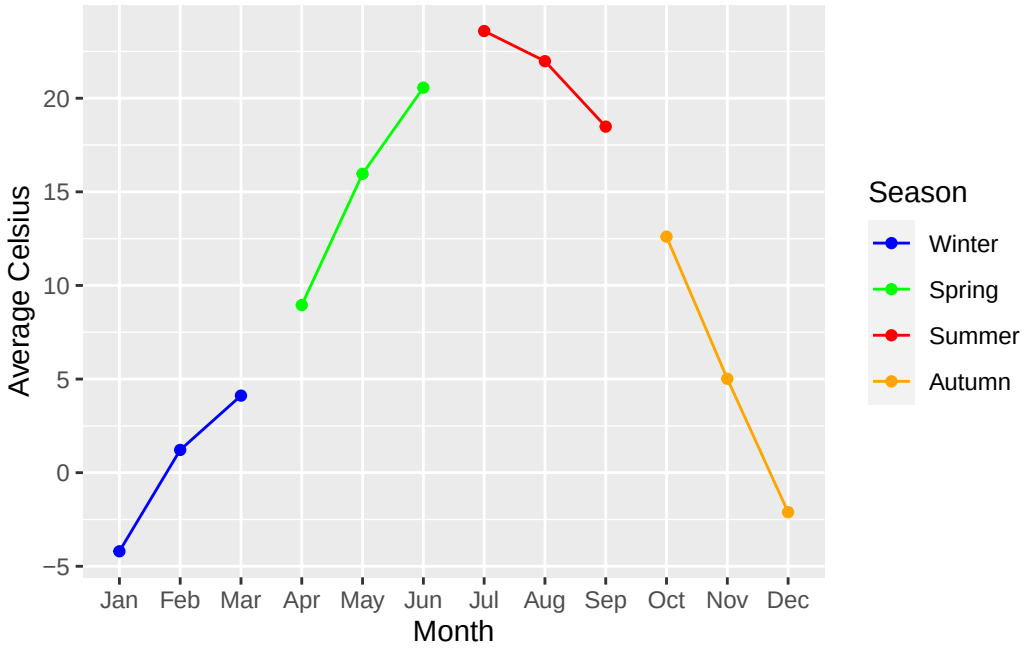
a.

```
nnmaps %>%
  mutate(tempc = 5/9*(temp - 32),
         month = fct(month,
                     levels = c("Jan", "Feb", "Mar", "Apr",
                                "May", "Jun", "Jul", "Aug",
                                "Sep", "Oct", "Nov", "Dec")),
         season = fct(season,
                      levels = c("Winter", "Spring"
                                 ,"Summer", "Autumn"))) -> nnmaps

nnmaps %>%
  group_by(month) %>%
  summarize(meantempc = mean(tempc),
            meano3 = mean(o3),
            meanpm10 = mean(pm10, na.rm = TRUE),
            meandewpoint = mean(dewpoint),
            season = first(season)) %>%
  ungroup() -> nnmaps_monthly

nnmaps_monthly %>% ggplot(aes(x = month, y = meantempc,
```

```
                                group = season, color = season)) +
  geom_point() +
  geom_line() +
  scale_color_manual(values = c("blue", "green", "red", "orange"),
                     name = "Season") +
  xlab("Month") + ylab("Average Celsius")
```

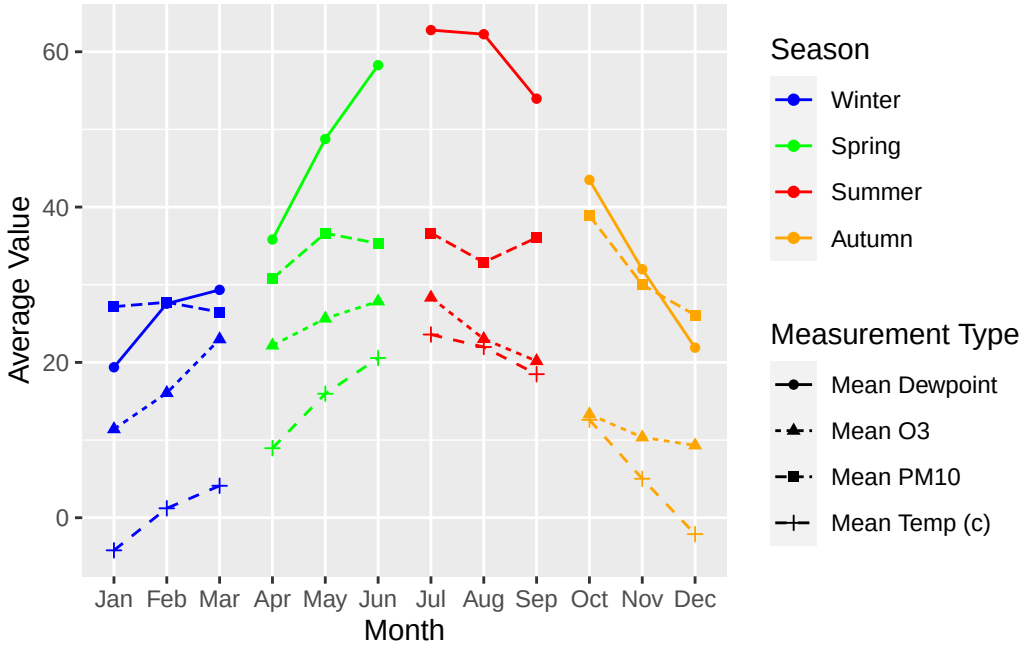


b.

```
nnmaps_monthly %>%
  gather(key = "measurement", value = "value",
         meantempc, meandewpoint,
         meanpm10, meano3) %>%
  mutate(measurement = replace(measurement,
                               measurement == "meandewpoint",
                               "Mean Dewpoint"),
         measurement = replace(measurement,
                               measurement == "meano3",
                               "Mean O3"),
         measurement = replace(measurement,
                               measurement == "meanpm10",
```

```
                                "Mean PM10"),
         measurement = replace(measurement,
                                measurement == "meantempc",
                                "Mean Temp (c)"),)-> nnmaps_monthly_tall

nnmaps_monthly_tall %>%
  ggplot(aes(x = month, y = value, color = season, group = interaction(season, measurement
  geom_point(aes(shape = measurement)) +
  geom_line(aes(linetype = measurement))  +
  scale_color_manual(values = c("blue", "green", "red", "orange"),
                     name = "Season") +
  scale_linetype(name = "Measurement Type") +
  scale_shape(name = "Measurement Type") +
  xlab("Month") + ylab("Average Value")
```



From looking at the plot, PM10 appears to have the least seasonality. You could also argue that O3 has weak seasonality too. Both temperature (obviously) and dewpoint have strong seasonal trends.

Problem 2 - OOP Programming

a.

```
setClass("poly",
         contains = "numeric",
         slots = c(powers = "numeric"))

setValidity("poly", function(object) {
  if (length(object) != length(object@powers)) {
    stop("Powers and coefficients of differing lengths")
  }
  if (any(is.na(object@powers))) {
    stop("Powers must not be missing")
  }
  if (any(is.na(object))) {
    stop("Coefficients must be numeric")
  }
  if (any(object@.Data == 0)) {
    stop("0 coefficients should not be stored")
  }
  if (any(diff(object@powers) > 0)) {
    stop("powers must be sorted in descending order")
  }
  TRUE
})
```

```
Class "poly" [in ".GlobalEnv"]

Slots:

Name:    .Data  powers
Class: numeric numeric

Extends:
Class "numeric", from data part
Class "vector", by class "numeric", distance 2
```

```
##' @title Create a `poly` object
##' @param lst A named list containing the coefficients, named for the power of
##'   the indeterminate.
```

```r
##' @return
##' @export
make_poly <- function(coeffs) {
  # Ensure input is ordered
  coeffs <- coeffs[sort(names(coeffs), decreasing = TRUE)]

  powers <- as.numeric(names(coeffs))
  values <- Reduce(c, coeffs)

  # Don't bother storing any 0's
  zeros <- which(values != 0)
  values <- values[zeros]
  powers <- powers[zeros]

  return(new("poly", values, powers = powers))
}

setMethod("show", "poly",
  function(object) {

    if (length(object@.Data) == 0) {
      # Short circuit on an all-zero coefficient poly
      cat(0, "\n")
    }
    # Generate each term by combining coefficient and power to
    # create terms of the form `cx^p`.
    terms <- paste0(object@.Data, "x^", object@powers)

    ### Special cases - to make it look nicer:
    # 1. x^0 is 1
    terms <- gsub("x\\^0$", "", terms)

    # 2. x^1 is x
    terms <- gsub("x\\^1$", "x", terms)

    # 3. Any coefficients of "1" can be dropped, *except* for the constant:
    terms <- gsub("^1x", "x", terms)
    terms <- gsub("^-1x", "-x", terms)

    ### Add signs
    # 1. Move `-` over to make spacing nicer - but NOT for the first entry
```

```
    terms[-1] <- gsub("^-", "- ", terms[-1])

    # 2. Add +'s if not having -'s, again excluding
    pluses <- !grepl("^-", terms)
    pluses[1] <- FALSE
    terms[pluses] <- paste("+", terms[pluses])

    cat(paste(terms, collapse = " "), "\n")
    return(invisible(object))
  }
)

setMethod("+", signature(e1 = "poly",
                         e2 = "poly"),
{
  function(e1, e2) {
    # Get list of all powers that exist in either object
    newpowers = unique(c(e1@powers, e2@powers))

    val <- c()
    pow <- c()

    for (p in newpowers) {
      # For each object, obtain the coefficient value,
      # or 0 if the power isn't found
      e1_p <- e1@powers == p
      e1v <- ifelse(any(e1_p), e1@.Data[e1_p], 0)
      e2_p <- e2@powers == p
      e2v <- ifelse(any(e2_p), e2@.Data[e2_p], 0)
      val <- c(e1v + e2v, val)
      pow <- c(p, pow)
    }

    # Strip any 0 coefficient terms
    zeros <- which(val != 0)
    val <- val[zeros]
    pow <- pow[zeros]

    # Re-order
    order <- order(pow, decreasing = TRUE)
    val <- val[order]
```

```
    pow <- pow[order]

    return(new("poly", val, powers = pow))
  }
})

setMethod("-", signature(e1 = "poly",
                         e2 = "poly"),
{
  function(e1, e2) {
    # (a + b) = (a + (-b))
    e2@.Data <- -1*e2@.Data
    return(e1 + e2)
  }
})
```

b.

```
p1 <- make_poly(list("2" = 3, "0" = 2))
p2 <- make_poly(list("3" = 7, "2" = -2, "1" = -1, "0" = 17))
p1
```

```
3x^2 + 2
```

```
p2
```

```
7x^3 - 2x^2 - x + 17
```

```
p1 + p2
```

```
7x^3 + x^2 - x + 19
```

```
p1 - p2
```

```
-7x^3 + 5x^2 + x - 15
```

For the very minor extra credit, my approach supports both negative and non-integer coefficients without complaint:

```
p3 <- make_poly(list("-3.5" = 7))
p4 <- make_poly(list(".2" = 3, "-3.5" = -2))
p3
```

```
7x^-3.5
```

```
p4
```

```
3x^0.2 - 2x^-3.5
```

```
p3 + p4
```

```
3x^0.2 + 5x^-3.5
```

```
p3 - p4
```

```
-3x^0.2 + 9x^-3.5
```

Problem 3 - data.table

```
library(data.table)
library(nycflights13)
```

a.

```
# Departure
flights <- data.table(flights)
merged <- merge(flights[, faa := origin],
                airports,
                by = "faa",
                all.x = TRUE)
merged[, .(N = .N,
           mean_delay = mean(dep_delay, na.rm = TRUE),
           med_delay = median(dep_delay, na.rm = TRUE)),
       by = name] |>
  _[N >= 10, !"N"] |>
```

```
  _[order(mean_delay, decreasing = TRUE)]
```

```
                 name mean_delay med_delay
1: Newark Liberty Intl   15.10795        -1
2: John F Kennedy Intl   12.11216        -1
3:          La Guardia   10.34688        -3
```

```
# Arrivals
flights <- data.table(flights)
merged <- merge(flights[, faa := dest],
                airports,
                by = "faa",
                all.x = TRUE)
# Above we grouped by name, which worked fine because all three origins have
# names. However, there are 3 destinations which don't have names. We need to
# group by `faa` to maintain those three destinations, so we'll save the `name`,
# though we need to keep the `faa` designation for the three airports without
# names
merged[, .(name = ifelse(is.na(first(name)), first(faa), first(name)),
           N = .N,
           mean_delay = mean(arr_delay, na.rm = TRUE),
           med_delay = median(arr_delay, na.rm = TRUE)),
       by = faa] |>
  _[N >= 10, !c("faa", "N")] |>
  _[order(mean_delay, decreasing = TRUE)] |>
  print(x = _, nrows = 10000)
```

```
                                    name   mean_delay med_delay
 1:                Columbia Metropolitan  41.76415094      28.0
 2:                           Tulsa Intl  33.65986395      14.0
 3:                  Will Rogers World  30.61904762      16.0
 4:                Jackson Hole Airport  28.09523810      15.0
 5:                       Mc Ghee Tyson  24.06920415       2.0
 6:               Dane Co Rgnl Truax Fld  20.19604317       1.0
 7:                           Richmond Intl  20.11125320       1.0
 8:        Akron Canton Regional Airport  19.69833729       3.0
 9:                       Des Moines Intl  19.00573614       0.0
10:                   Gerald R Ford Intl  18.18956044       1.0
11:                      Birmingham Intl  16.87732342      -2.0
12:         Theodore Francis Green State  16.23463687       1.0
```

```
13: Greenville-Spartanburg International  15.93544304      -0.5
14:    Cincinnati Northern Kentucky Intl  15.36456376      -3.0
15:            Savannah Hilton Head Intl  15.12950601      -1.0
16:          Manchester Regional Airport  14.78755365      -3.0
17:                          Eppley Afld  14.69889841      -2.0
18:                               Yeager  14.67164179      -1.5
19:                    Kansas City Intl  14.51405836       0.0
20:                          Albany Intl  14.39712919      -4.0
21:                General Mitchell Intl  14.16722038       0.0
22:                       Piedmont Triad  14.11260054      -2.0
23:                Washington Dulles Intl  13.86420212      -3.0
24:                Cherry Capital Airport  12.96842105     -10.0
25:               James M Cox Dayton Intl  12.68048606      -3.0
26:    Louisville International Airport  12.66938406      -2.0
27:                  Chicago Midway Intl  12.36422360      -1.0
28:                      Sacramento Intl  12.10992908       4.0
29:                    Jacksonville Intl  11.84483416      -2.0
30:                       Nashville Intl  11.81245891      -2.0
31:                Portland Intl Jetport  11.66040210      -4.0
32:                Greater Rochester Intl  11.56064461      -5.0
33:     Hartsfield Jackson Atlanta Intl  11.30011285      -1.0
34:                Lambert St Louis Intl  11.07846451      -3.0
35:                         Norfolk Intl  10.94909344      -4.0
36:            Baltimore Washington Intl  10.72673385      -5.0
37:                         Memphis Intl  10.64531435      -2.5
38:                   Port Columbus Intl  10.60132291      -3.0
39:                  Charleston Afb Intl  10.59296847      -4.0
40:                    Philadelphia Intl  10.12719014      -3.0
41:                  Raleigh Durham Intl  10.05238095      -3.0
42:                    Indianapolis Intl   9.94043412      -3.0
43:            Charlottesville-Albemarle   9.50000000      -5.0
44:               Cleveland Hopkins Intl   9.18161129      -5.0
45:       Ronald Reagan Washington Natl   9.06695204      -2.0
46:                       Burlington Intl   8.95099602      -4.0
47:                  Buffalo Niagara Intl   8.94595186      -5.0
48:                Syracuse Hancock Intl   8.90392501      -5.0
49:                           Denver Intl   8.60650021      -2.0
50:                       Palm Beach Intl   8.56297210      -3.0
51:                                  BQN   8.24549550      -1.0
52:                              Bob Hope   8.17567568      -3.0
53:      Fort Lauderdale Hollywood Intl   8.08212154      -3.0
54:                           Bangor Intl   8.02793296      -9.0
55:           Asheville Regional Airport   8.00383142      -1.0
```

```
56:                                PSE   7.87150838      0.0
57:                  Pittsburgh Intl   7.68099053     -5.0
58:                   Gallatin Field   7.60000000     -2.0
59:               NW Arkansas Regional   7.46572581     -2.0
60:                        Tampa Intl   7.40852503     -4.0
61:             Charlotte Douglas Intl   7.36031885     -3.0
62:          Minneapolis St Paul Intl   7.27016886     -5.0
63:                  William P Hobby   7.17618819     -4.0
64:                     Bradley Intl   7.04854369    -10.0
65:                  San Antonio Intl   6.94537178     -9.0
66:                  South Bend Rgnl   6.50000000     -3.5
67:   Louis Armstrong New Orleans Intl   6.49017497     -6.0
68:                     Key West Intl   6.35294118      7.0
69:                     Eagle Co Rgnl   6.30434783     -4.0
70:              Austin Bergstrom Intl   6.01990875     -5.0
71:                 Chicago Ohare Intl   5.87661475     -8.0
72:                        Orlando Intl   5.45464309     -5.0
73:             Detroit Metro Wayne Co   5.42996346     -7.0
74:                     Portland Intl   5.14157973     -5.0
75:                     Nantucket Mem   4.85227273     -3.0
76:                  Wilmington Intl   4.63551402     -7.0
77:                 Myrtle Beach Intl   4.60344828    -13.0
78:  Albuquerque International Sunport   4.38188976     -5.5
79:       George Bush Intercontinental   4.24079040     -5.0
80:     Norman Y Mineta San Jose Intl   3.44817073     -7.0
81:             Southwest Florida Intl   3.23814963     -5.0
82:                      San Diego Intl   3.13916574     -5.0
83:           Sarasota Bradenton Intl   3.08243131     -5.0
84:         Metropolitan Oakland Intl   3.07766990     -9.0
85: General Edward Lawrence Logan Intl   2.91439222     -9.0
86:                 San Francisco Intl   2.67289152     -8.0
87:                               SJU   2.52052659     -6.0
88:                      Yampa Valley   2.14285714      2.0
89:            Phoenix Sky Harbor Intl   2.09704733     -6.0
90:          Montrose Regional Airport   1.78571429    -10.5
91:                  Los Angeles Intl   0.54711094     -7.0
92:             Dallas Fort Worth Intl   0.32212685     -9.0
93:                        Miami Intl   0.29905978     -9.0
94:                   Mc Carran Intl   0.25772849     -8.0
95:               Salt Lake City Intl   0.17625459     -8.0
96:                        Long Beach  -0.06202723    -10.0
97:             Martha\\\\'s Vineyard  -0.28571429    -11.0
98:               Seattle Tacoma Intl  -1.09909910    -11.0
```

```
 99:                         Honolulu Intl  -1.36519258      -7.0
100:                                  STT  -3.83590734      -9.0
101:             John Wayne Arpt Orange Co  -7.86822660     -11.0
102:                     Palm Springs Intl -12.72222222     -13.5
                                     name   mean_delay med_delay
```

b.

```
planes <- data.table(planes)
merged <- merge(flights,
                planes,
                by = "tailnum",
                all.x = TRUE)
allmodels <- merged[, `:=`(nflights = .N,
                           avgmph = mean(distance/(air_time/60), na.rm = TRUE)),
                    by = model]

allmodels[allmodels[, .I[which.max(avgmph)]],.(model, avgmph, nflights)]
```

```
    model   avgmph nflights
1: 777-222 482.6254        4
```