Statistical Computing 1, Stat 590 Homework 20 Name: Answer Key
Fall 2015 Prof. Erik B. Erhardt

Part I. (30 points) Do all calculations in LaTeX + R + knitr. For this assignment, all R code should well commented and be visible (`echo=TRUE`) in the document where you have written it. Every time you create or modify an object, please show the results with the appropriate function. Please do not display complete objects when they are large.

Weather data: This is an exercise in manipulating data. We will be using daily weather data from the Albuquerque International Airport (KABQ).

This is a continuation of HW03.

(30pts) **1. Apply, summarize, and test**

(a) (10 pts) Use `ddply()` to find the 5-number summary by year for each variable, as well as the number of samples per year. Is it easier to do this on the long-form or wide-form data?

Solution:

```
## Code from HW03

fn.list <- c(
    "http://www.wunderground.com/history/airport/KABQ/2009/1/1/CustomHistory.html?dayend=31&monthend=12&yearend=2009&req_city=NA&req_state=NA&req_statename=N
  , "http://www.wunderground.com/history/airport/KABQ/2010/1/1/CustomHistory.html?dayend=31&monthend=12&yearend=2010&req_city=NA&req_state=NA&req_statename=N
  , "http://www.wunderground.com/history/airport/KABQ/2011/1/1/CustomHistory.html?dayend=31&monthend=12&yearend=2011&req_city=NA&req_state=NA&req_statename=N
  , "http://www.wunderground.com/history/airport/KABQ/2012/1/1/CustomHistory.html?dayend=31&monthend=12&yearend=2012&req_city=NA&req_state=NA&req_statename=N
  , "http://www.wunderground.com/history/airport/KABQ/2013/1/1/CustomHistory.html?dayend=31&monthend=12&yearend=2013&req_city=NA&req_state=NA&req_statename=N
  , "http://www.wunderground.com/history/airport/KABQ/2014/1/1/CustomHistory.html?dayend=31&monthend=12&yearend=2014&req_city=NA&req_state=NA&req_statename=N
  , "http://www.wunderground.com/history/airport/KABQ/2015/1/1/CustomHistory.html?dayend=31&monthend=12&yearend=2015&req_city=NA&req_state=NA&req_statename=N
  )

# list to hold all the data files
all.dat <- as.list(new.env())

# read all data
for (i.dat in 1:length(fn.list)) {
  # read each year's data
  dat.temp <- read.csv( fn.list[i.dat]
                      , stringsAsFactors = FALSE
                      )
  # put data into it's own list
  all.dat[[i.dat]] <- dat.temp
}

# all the column names match except for the last data.frame
colnames(all.dat[[1]]) == colnames(all.dat[[7]])
##  [1] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [12]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [23]  TRUE
# change that column name to match the others
colnames(all.dat[[7]])[1] <- "MST"

# combine all data together into one data.frame
dat <- do.call("rbind", all.dat)

#str(dat)

library(lubridate)
dat$Date <- ymd(dat$MST)
#str(dat[, c("MST", "Date")])

library(stringr)
# the result of spliting the strings is a list of lists
dat.ymd <- str_split(dat$MST, "-")
#head(dat.ymd)

# bind the lists together using rbind
dat.ymd <- do.call("rbind", str_split(dat$MST, "-"))
#head(dat.ymd)
dat.ymd <- data.frame(Year = dat.ymd[,1], Month = dat.ymd[,2], stringsAsFactors = FALSE)
#str(dat.ymd)

# combine all data together into one data.frame
dat <- cbind(dat, dat.ymd)
```

```
# order the date factor variables
dat$Year <- factor(dat$Year)
dat$Month <- factor(dat$Month, levels = 1:12, labels = as.character(1:12), ordered = TRUE)

# subset data
dat.sub <- subset(dat, select = c("Mean.TemperatureF"
                                , "Mean.Wind.SpeedMPH"
                                , "PrecipitationIn"
                                , "Date"
                                , "Year"
                                , "Month"
                                ))
#str(dat.sub)

library(reshape2)
dat.sub.long <- melt(dat.sub, id.vars = c("Date", "Year", "Month"))
#str(dat.sub.long)
```

Easier to do in long format data. Leap years have correct number of observations, as well as last partial last year.
Not a lot of rain, just what I like in Albuquerque! 2011 was the super cold year when house water pipes froze and burst. Otherwise, pretty consistent over the last several years.

```
library(plyr)
dat.5sum <- ddply(dat.sub.long
                , c("variable", "Year")
                , function(.X) {
                    # calculate fivenum summary and make as a row vector
                    fns <- matrix(fivenum(.X$value), nrow = 1)
                    # name the columns
                    colnames(fns) <- c("Min", "Q1", "M", "Q3", "Max")
                    # convert to a data.frame
                    out <- as.data.frame(fns)
                    # number of samples per year
                    out <- cbind(out, data.frame(n = length(.X$value)))
                    return(out)
                  }
                  )
```

```
## Error in x[floor(d)] + x[ceiling(d)]:  non-numeric argument to binary operator
```

```
dat.5sum
```

```
## Error in eval(expr, envir, enclos):  object 'dat.5sum' not found
```

(b) (10 pts) Use `subset()` and `ddply()` to find the 5-number summary of max wind and mean temp by month.
Solution:

```
# subset data
dat.sub2 <- subset(dat, select = c("Mean.TemperatureF"
                                 ,  "Max.Wind.SpeedMPH"
                                 , "Date"
                                 , "Year"
                                 , "Month"
                                 ))
str(dat.sub2)
```

```
## 'data.frame': 2441 obs. of  5 variables:
##  $ Mean.TemperatureF: int  42 43 44 34 31 34 36 43 41 37 ...
##  $ Max.Wind.SpeedMPH: int  10 9 22 36 22 25 22 14 23 18 ...
##  $ Date             : POSIXct, format: "2009-01-01" ...
##  $ Year             : Factor w/ 7 levels "2009","2010",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
##  $ Month           : Ord.factor w/ 12 levels "1"<"2"<"3"<"4"<..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
library(reshape2)
dat.sub2.long <- melt(dat.sub2, id.vars = c("Date", "Year", "Month"))
str(dat.sub2.long)
```

```
## 'data.frame': 4882 obs. of  5 variables:
##  $ Date    : POSIXct, format: "2009-01-01" ...
##  $ Year    : Factor w/ 7 levels "2009","2010",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Month   : Ord.factor w/ 12 levels "1"<"2"<"3"<"4"<..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ variable: Factor w/ 2 levels "Mean.TemperatureF",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ value   : int  42 43 44 34 31 34 36 43 41 37 ...
```

```
library(plyr)
dat.5sum.mean <- ddply(dat.sub2.long
               , c("Year", "Month")
               , function(.X) {
                   ## Max.Wind.SpeedMPH
                   # calculate fivenum summary and make as a row vector
                   fns <- matrix(fivenum(subset(.X, variable == "Max.Wind.SpeedMPH")$value), nrow = 1)
                   # name the columns
                   colnames(fns) <- c("MaxWind_Min", "MaxWind_Q1", "MaxWind_M", "MaxWind_Q3", "MaxWind_Max")
                   # convert to a data.frame
                   fns <- as.data.frame(fns)

                   ## Mean.TemperatureF
                   MeanTemp <- mean(subset(.X, variable == "Mean.TemperatureF")$value)

                   # number of samples per month
                   out <- cbind(fns, MeanTemp)
                   return(out)
                 }
                 )
head(dat.5sum.mean, 10)
```

```
##    Year Month MaxWind_Min MaxWind_Q1 MaxWind_M MaxWind_Q3 MaxWind_Max
## 1  2009     1           7       13.5      17.0       23.0          36
## 2  2009     2          12       15.5      20.5       28.5          38
## 3  2009     3           8       18.0      24.0       30.5          43
## 4  2009     4          10       18.0      24.0       35.0          51
## 5  2009     5          10       22.0      26.0       35.0          45
## 6  2009     6          14       22.0      26.5       31.0          40
## 7  2009     7          10       21.0      23.0       29.0          38
## 8  2009     8           8       17.5      23.0       29.5          39
## 9  2009     9          12       16.0      22.0       29.0          38
## 10 2009    10          10       14.5      22.0       24.0          36
##    MeanTemp
## 1  40.80645
## 2  44.82143
## 3  50.48387
## 4  55.96667
## 5  69.03226
## 6  73.36667
## 7  80.61290
## 8  77.70968
## 9  68.60000
## 10 56.29032
```

(c) (10 pts) Propose and test a weather-related hypothesis using this dataset. Use the result with `\Sexpr{}` to refer to a statistic in a sentence to interpret the result.

Solution: Are summers getting warmer?
Use regression and test for increasing slope of August temperature over years.

```
# subset data
dat.hyp <- subset(dat.5sum.mean, Month == "8", select = c("Year", "MeanTemp"))
```

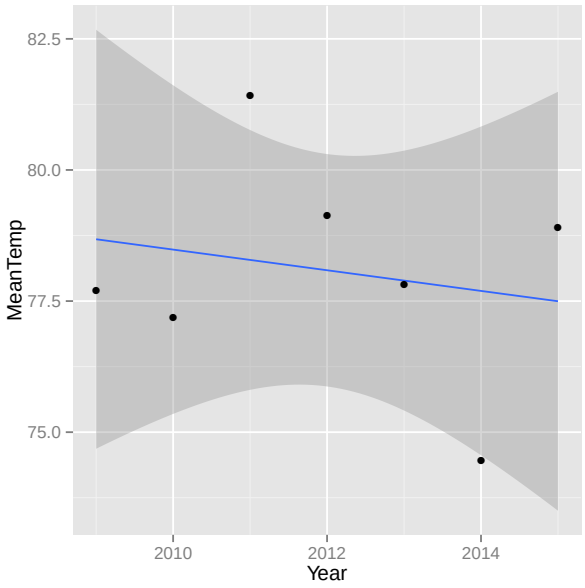
```
# numeric year
dat.hyp$Year <- as.numeric(as.character(dat.hyp$Year))

library(ggplot2)
p <- ggplot(dat.hyp, aes(x = Year, y = MeanTemp))
p <- p + stat_smooth(method = lm)
p <- p + geom_point()
print(p)

lm.fit <- lm(Year ~ MeanTemp, data = dat.hyp)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = Year ~ MeanTemp, data = dat.hyp)
##
## Residuals:
##       8      20      32      44      56      68      80
## -3.0769 -2.1820 -0.3216  0.2121  0.9428  1.2597  3.1661
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2027.8992    34.7939  58.283 2.81e-08 ***
## MeanTemp      -0.2036     0.4454  -0.457    0.667
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.318 on 5 degrees of freedom
## Multiple R-squared:  0.04011,Adjusted R-squared:  -0.1519
## F-statistic: 0.2089 on 1 and 5 DF,  p-value: 0.6668
```

```
# pull values from the coefficient table
p.val <- coef(summary(lm.fit))[2,4]
```



Because the p-value $= 0.6668 < 0.05$, we fail to reject the null hypothesis; we have

insufficient evidience that summers are getting warmer.