Lottery Case Study
Note: R commands are in bold
Ala Abdelbaki

1. Save lottery data
   a. name “lottery1”
   b. can save as plain text (txt) format

2. R needs to read the data set
   
   ```
   lot<-read.table("lottery1.txt")
   ```
   a. “Lot” used so that it is defined
   b. Read.table used for files saved in txt format
   c. NA refers to missing values, different from not possible or null
   d. A value cannot be compared to NA (rather than an answer, NA will be given)

3. Convert data set into matrix
   a. Matrix is a rectangular grid of data; all must be of the same type (numbers, characters, etc.)
   b. Lot[,1] to list the data in the first column (days of the month)
   c. Lot[1,] to list the data in the first row (months of the year)
   
   ```
   lot<-as.matrix(lot)
   ```
   d. “as.” is used to convert one data type to another(e.g.data to matrix, vector, data frame)
   e. Data set now has row numbers

4. Convert data set into vector
   
   ```
   lot<-as.vector(lot)
   ```
   a. “Vector” produces all data in a single row
   b. All values must have the same type
   c. When testing if it is a vector by using lot[1, ] and lot[, 1], an error message will appear since the specified dimensions don’t exist in a vector (i.e. rows and columns don’t exist in a single vector). Rather, these commands are characteristically used in a matrix.
   d. But, for example, lot[1] returns the value 305.

5. Remove any missing values from the data set
   
   ```
   lot<-lot[!is.na(lot)]
   ```
   a. ! refers to the negation (if value is NA, then remove)
   b. All missing values (NA) are now removed and there are 366 values in the data set
6. Plot the graph using linear regression

```r
data <- 1:366
plot(date, lot)
cor(date, lot)

# a. Modified data set with missing values is used
# b. Date is on the x-axis, lot is on the y-axis
# c. slope => negative (what does this indicate?)

fit <- lm(lot ~ date)
abline(fit)
```

d. Lm= linear model function
   i. (y ~ x) Read as “lot (y) is modeled by date (x)”
   ii. Model assumes an intercept term and a linear model
e. “abline” is used to add the regression line to the scatterplot
   i. Line of best fit

7. Parameters

```r
par(mfrow = c(2,2))
```

a. Parameters of the graph can be set using the par function
b. Most commonly used is mfrow (multiple figures by row) and mfcol (multiple figures by column) which determine the layout of multiple graphs in one graphing device
c. By specifying mfrow = c(n,m) the results will be an n*m matrix of graphs that are filled by rows

8. Repeated numbers

a. Rep() function is used
b. Replicates the values in x in a vector
c. Example: `rep(1,3)` will display `[1] 1 1 1 (the number 1 three times)

9. Looping

a. For () loops
   i. Basic structure is: for (varname in seq) {statement(s)}

```r
for (i in 1:366) rmean[i] <- mean(lot[(max(1,i-1):min(366,i+1))])
```

ii. Varname is the variable i and the vector is the set of numbers 1 through 366
iii. The statements are evaluated with the given value for varname
iv. Running mean of i ...
v. The loop runs over the values of the vector 1:366
vi. Conclusion: graphs vary by length of running mean

10. Trend Lines

a. Produces a smooth curve summarizing the relationship between the two variables in a scatterplot
b. Smooth spline: fits the data using cubic splines
i. df: the desired equivalent number of degrees of freedom (trace of the smoother matrix)

ii. spar: smoothing parameter

iii. lambda: the value of $\lambda$ corresponding to spar

\[
\text{fit} \gets \text{smooth.spline(lot~date,df=10)}
\]

c. This produces a smoothing curve which suggests the relationship between lot and date, with a degree of freedom of 10 (i.e. lot as a function of date)

d. In summary: the lower the value of the df, the greater the smoothing (which corresponds to a larger bandwidth)

11. Spearman Rank Correlation

a. A non-parametric test devised to assess the relationship between two variables that are not linear.

b. Non-parametric tests are especially useful with small data sets that have a natural ranking order. Since the data sets are small, we usually do not have knowledge about the population distribution(s) at hand.

\[
\text{ht} \gets \text{c}(73, 71, 72, 173/2.54, 67, 72, 72)
\]

c. The data are ordered from smallest to largest; ties are averaged.

\[
\text{rht} \gets \text{rank(ht)}
\]

rht
d. This means to rank the data set for height

\[
\text{rank.correlation} \gets \text{cor(rht,rwt)}
\]

rank.correlation
e. This is used to find the rank correlation between height and weight

\[
\text{cor(wt,ht,method="spearman")}
\]

f. Defines method as Spearman

\[
\text{cor.test(wt,ht,alternative="greater",method="spearman")}
\]

g. Greater indicates that the association between height and weight is positive

h. In R, put in “cor.test” in help for the proper syntax

i. $H_0$: weight and height do not jointly increase

j. $H_a$: weight and height jointly increase

12. Hypothesis Testing on Spearman Rank Correlation

a. \[
\text{cor.test(date,lot,alternative="less",method="spearman")}
\]

b. “Less” indicates that the association between height and weight is negative (i.e. as birthdate increases, lottery number decreases)