Homework 3

Angela Zhang

10/12/2017

Getting started with Tidyverse

(a)

```
apply(faithful,2,mean)
```

```
## eruptions   waiting
##  3.487783 70.897059
```

(b)

```
library(magrittr)

faithful %>%
  apply(2,mean)
```

```
## eruptions   waiting
##  3.487783 70.897059
```

(c)

```
library(tidyverse)
```

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr
```

```
## Conflicts with tidy packages ----------------------------------------------
```

```
## filter(): dplyr, stats
## lag():    dplyr, stats
```

```
faithful %>%
  summarize(mean = mean(eruptions))
```

```
##       mean
## 1 3.487783
```

(d)

```
faithful %>%
  summarize_all(mean)
```

```
##   eruptions  waiting
## 1  3.487783 70.89706
```

(e)

```
dim(faithful)
```

```
## [1] 272   2
```

```
as.tibble(faithful)
```

```
## # A tibble: 272 x 2
##    eruptions waiting
##  *     <dbl>   <dbl>
##  1     3.600      79
##  2     1.800      54
##  3     3.333      74
##  4     2.283      62
##  5     4.533      85
##  6     2.883      55
##  7     4.700      88
##  8     3.600      85
##  9     1.950      51
## 10     4.350      85
## # ... with 262 more rows
```

Intuitive data exploration

(a) Subset is part of base R and doesn't discard row names

(b)

```
airquality %>%
  filter(Temp > 85) %>%
  summarize(mean = mean(Ozone, na.rm = T), sd = sd(Ozone, na.rm = T))
```

```
##       mean       sd
## 1 79.22222 20.89964
```

(c)

```
airquality %>%
  group_by(Temp > 85) %>%
  summarize(mean = mean(Ozone, na.rm = T), sd = sd(Ozone, na.rm = T))
```

```
## # A tibble: 2 x 3
##   `Temp > 85`     mean       sd
##         <lgl>    <dbl>    <dbl>
## 1       FALSE 30.87640 27.25462
## 2        TRUE 79.22222 20.89964
```

```
## using by() (from lecture 2)
by(airquality$Ozone, list(month = airquality$Month), mean, na.rm = TRUE)
```

```
## month: 5
## [1] 23.61538
## --------------------------------------------------------
## month: 6
## [1] 29.44444
## --------------------------------------------------------
## month: 7
## [1] 59.11538
## --------------------------------------------------------
## month: 8
## [1] 59.96154
## ---------------------------------------------------------
## month: 9
## [1] 31.44828
```

Function composition

(a)

`f <- . %>% cos %>% sin` creates the function `sin(cos(x))` where the user provides the `x`. For example:

```
f <- . %>% cos %>% sin

f(2)
```

```
## [1] -0.4042392
```

```
sin(cos(2))
```

```
## [1] -0.4042392
```

(b)

The compound pipe operator `%<>%` is used as shorthand notation in cases where you overwrite a variable. For example, I used instead of using `beaver1$temp <- sqrt(beaver1$temp)`,you can use the command `beaver1 %<>% sqrt` to achieve the same effect of finding the square root of all the temperatures in the `beaver1` dataset.

(c)

I wrote a function that converts Fahrenheit to Celsius and then I used that function with the compound pipe operator

```
head(airquality$Temp)
```

```
## [1] 67 72 74 62 56 66
```

```
FtoC <- function(x){round((x-32)*(5/9),0)}

airquality$Temp %<>% (function(x){round((x-32)*(5/9),0)})

head(airquality$Temp)
```

```
## [1] 19 22 23 17 13 19
```