



Exploratory Data Analysis

Part 2: Multivariate graphics, Numeric summaries

Math 445

March 30, 2016

Plotting multiple variables

Basic bivariate graphics

Variable type	Plot suggestions
Quantitative vs. quantitative	Scatterplot
Quantitative vs. categorical	Side-by-side boxplots Facetted histograms/densities

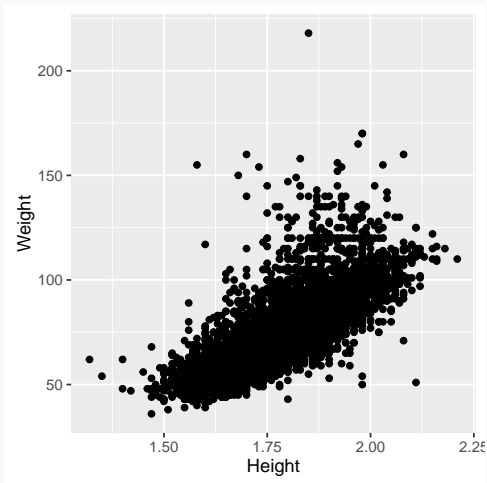
Data: 2012 Olympic Athletes

```
oly12 <- read.table("https://raw.githubusercontent.com/math445-LU/2016/master/data/oly12.csv",
  sep = ",", header = TRUE)
oly12$$Sport <- abbreviate(oly12$$Sport, 12)
str(oly12)
```

```
## 'data.frame': 10384 obs. of 14 variables:
## $ Name : Factor w/ 10366 levels "Aaron Brown",...: 5353 121 4117 16 6033 5686 6061 670
## $ Country: Factor w/ 205 levels "Afghanistan",...: 144 195 68 125 154 68 8 125 94 3 ...
## $ Age : int 23 33 30 24 26 27 30 23 27 19 ...
## $ Height : num 1.7 1.93 1.87 NA 1.78 1.82 1.82 1.87 1.9 1.7 ...
## $ Weight : int 60 125 76 NA 85 80 73 75 80 NA ...
## $ Sex : Factor w/ 2 levels "F","M": 2 2 2 2 1 2 1 2 2 2 ...
## $ DOB : Date, format: "1989-02-06" NA ...
## $ PlaceOB: Factor w/ 4108 levels "", "Aachen (GER)",...: 2486 3302 398 48 3436 1 1 1 117
## $ Gold : int 0 0 0 0 0 0 0 0 0 0 ...
## $ Silver : int 0 0 0 0 0 0 0 0 0 0 ...
## $ Bronze : int 0 0 0 0 0 0 0 0 0 0 ...
## $ Total : int 0 0 0 0 0 0 0 0 0 0 ...
## $ Sport : chr "Judo" "Athletics" "Athletics" "Boxing" ...
## $ Event : Factor w/ 763 levels "Group All-Around",...: 350 405 251 443 699 406 726 403
```

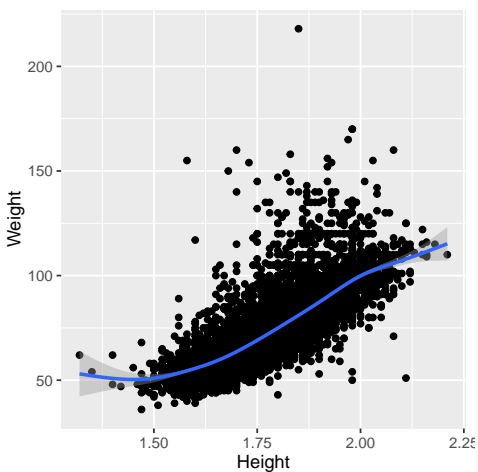
Scatterplots + Smoother

```
ggplot(data = oly12) +  
  geom_point(mapping = aes(x = Height, y = Weight))
```



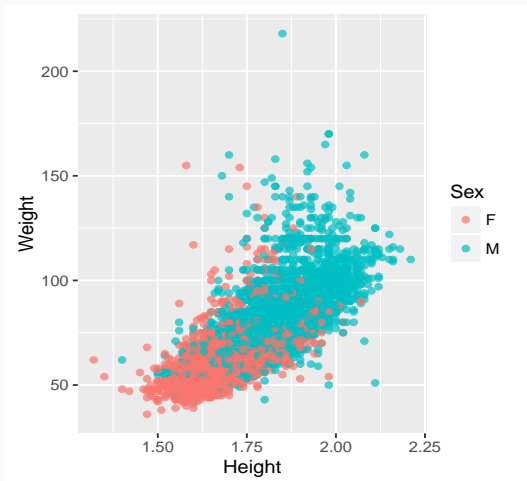
Scatterplots + Smoother

```
ggplot(data = oly12) +  
  geom_point(mapping = aes(x = Height, y = Weight)) +  
  geom_smooth(mapping = aes(x = Height, y = Weight))
```



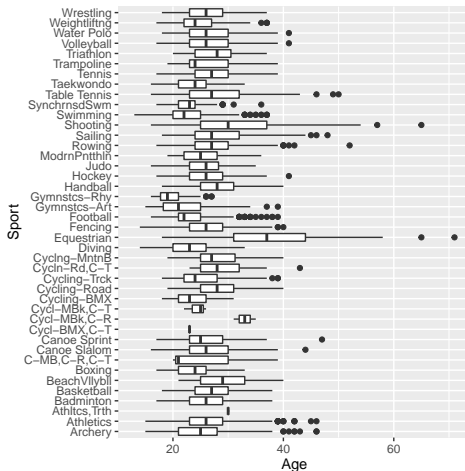
Scatterplots with extra variables

```
ggplot(data = oly12) +  
  geom_point(mapping = aes(x = Height, y = Weight, colour = Sex), alpha = 0.7)
```



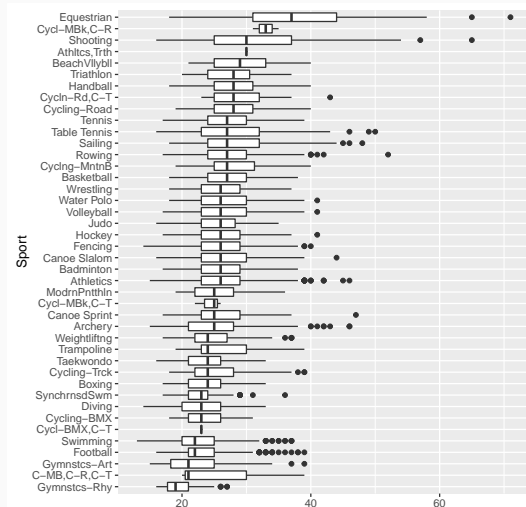
Side-by-side boxplots

```
ggplot(data = oly12) +  
  geom_boxplot(mapping = aes(x = Sport, y = Age)) +  
  coord_flip()
```



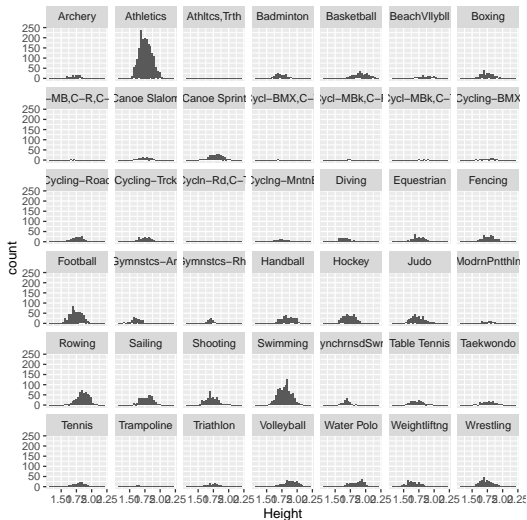
Side-by-side boxplots

```
ggplot(data = oly12) +  
  geom_boxplot(mapping = aes(x = reorder(Sport, Age, median), y = Age)) +  
  coord_flip() +  
  xlab("Sport")
```



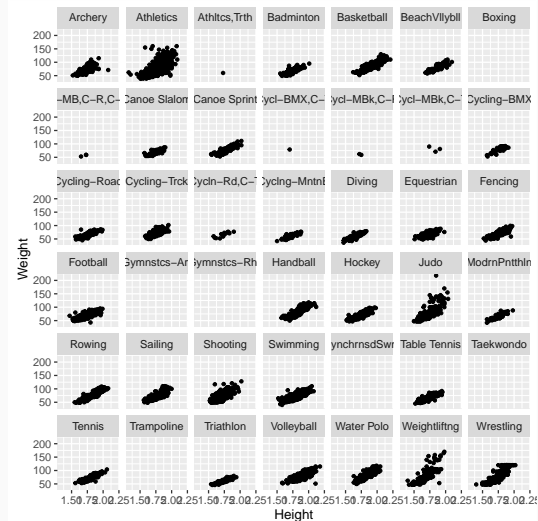
Facetting

```
ggplot(data = oly12) +  
  geom_histogram(mapping = aes(x = Height)) +  
  facet_wrap(~ Sport)
```



Facetting

```
ggplot(data = oly12) +  
  geom_point(mapping = aes(x = Height, y = Weight), size = 1) +  
  facet_wrap(~ Sport)
```



Summarizing data numerically

A common way to summarize a variable is to extract the column from the data frame and deal with it separately.

```
mean(oly12$Age)
median(oly12$Age)
sd(oly12$Age)
var(oly12$Age)
quantile(oly12$Age, probs = c(0.2, 0.4, 0.6, 0.8))
```

Summaries by group

To obtain summaries by group we can use functionality found in the `dplyr` package:

```
# install.packages('dplyr') # uncomment if not installed  
library(dplyr)
```

In addition to groupwise processing, `dplyr` provides **chaining syntax**:

```
# Regular syntax  
object_name <- function_name(data = data_table, arguments)  
  
# Chaining syntax  
object_name <-  
  data_table %>%  
  function_name(arguments)
```

Summaries by group

Suppose that we are interested in calculating the average age of 2012 Olympic athletes by sport:

```
age_sport <-  
  oly12 %>%  
  group_by(Sport) %>%  
  summarise(avgAge = mean(Age))  
  
head(age_sport)  
  
## Source: local data frame [6 x 2]  
##  
##       Sport    avgAge  
##       (chr)    (dbl)  
## 1    Archery 26.07438  
## 2  Athletics 26.17131  
## 3 Athltcs,Trth 30.00000  
## 4   Badminton 26.15663  
## 5   Basketball 27.17844  
## 6 BeachVllybll 29.18280
```

Summaries by group

We can also quickly obtain the medal count by country:

```
medal_count <-  
  oly12 %>%  
    group_by(Country) %>%  
    summarise(Gold = sum(Gold), Silver = sum(Silver), Bronze = sum(Bronze)) %>%  
    arrange(desc(Gold), desc(Silver), desc(Bronze))  
  
head(medal_count)
```

```
## Source: local data frame [6 x 4]
```

```
##
```

```
##           Country  Gold Silver Bronze
```

```
##           (fctr) (int)  (int)  (int)
```

```
## 1 United States of America    40     19     20
```

```
## 2 People's Republic of China    25     15     13
```

```
## 3                Germany    21     11      8
```

```
## 4           Great Britain    11     13     20
```

```
## 5                France    11     11     11
```

```
## 6       Republic of Korea    10      2     10
```


What if you only want summaries for one group?

- Create the summaries for all of the groups and then extract the group of interest
- Extract data for the group of interest and then create summaries

The `filter` command in the `dplyr` package allows you to easily subset a data frame:

```
filter(data, criteria)
```

Subsetting

```
oly12 %>%  
  filter(Country == "United States of America") %>%  
  group_by(Sex) %>%  
  summarise(avgAge = mean(Age))  
  
## Source: local data frame [2 x 2]  
##  
##      Sex    avgAge  
##   (fctr)    (dbl)  
## 1      F 26.44528  
## 2      M 27.73123
```

Subsetting

```
oly12 %>%  
  filter(Gold > 0 | Silver > 0 | Bronze > 0) %>%  
  ggplot() +  
  geom_density(mapping = aes(x = Age, group = Sex, fill = Sex), alpha = 0.7)
```

