Multivaraite Statistical Methods - Lab 01

Maximilian Pfundstein (maxpf364) and Hector and Aashana and Lakshidaa

2019-11-07

Contents

1 Describing Individual Variables

Consider the data set in the `T1-9.dat` file, National track records for women. For 55 different countries we have the national records for 7 variables (100, 200, 400, 800, 1500, 3000m and marathon). Use R to do the following analyses.

1.1 Describing the Variables

Task: Describe the 7 variables with mean values, standard deviations e.t.c.

Answer: First we import, name and look at the track times.

```r
track_times = read.table("data/T1-9.dat")
colnames(track_times) = c("country", "100m", "200m", "400m",
                          "800m", "1500m", "3000m", "marathon")
head(track_times)
```

```
##   country  100m  200m  400m 800m 1500m 3000m marathon
## 1     ARG 11.57 22.94 52.50 2.05  4.25  9.19   150.32
## 2     AUS 11.12 22.23 48.63 1.98  4.02  8.63   143.51
## 3     AUT 11.15 22.70 50.62 1.94  4.05  8.78   154.35
## 4     BEL 11.14 22.48 51.45 1.97  4.08  8.82   143.05
## 5     BER 11.46 23.05 53.30 2.07  4.29  9.81   174.18
## 6     BRA 11.17 22.60 50.62 1.97  4.17  9.04   147.41
```

We see that the times for the 100m, 200m and 400m are in seconds, whereas the times for 1500m, 3000m and the marathon are measured in minutes. So first we have to adjust that.

```r
track_times[,5:8] = track_times[,5:8] * 60
head(track_times)
```

```
##   country  100m  200m  400m  800m 1500m 3000m marathon
## 1     ARG 11.57 22.94 52.50 123.0 255.0 551.4   9019.2
## 2     AUS 11.12 22.23 48.63 118.8 241.2 517.8   8610.6
## 3     AUT 11.15 22.70 50.62 116.4 243.0 526.8   9261.0
## 4     BEL 11.14 22.48 51.45 118.2 244.8 529.2   8583.0
## 5     BER 11.46 23.05 53.30 124.2 257.4 588.6  10450.8
## 6     BRA 11.17 22.60 50.62 118.2 250.2 542.4   8844.6
```

```r
track_times_mean = apply(track_times[,2:8], 2, mean)
track_times_median = apply(track_times[,2:8], 2, median)
track_times_sd = apply(track_times[,2:8], 2, sd)

track_times_mean
```

```
##       100m       200m       400m       800m      1500m      3000m 
##   11.35778   23.11852   51.98907  121.34444  251.36667  544.84444 
##   marathon 
## 9217.15556
```

```r
track_times_median
```

```
##     100m     200m     400m     800m    1500m    3000m marathon 
##   11.325   22.980   51.645  120.300  246.000  530.700 8905.800
```

```r
track_times_sd
```

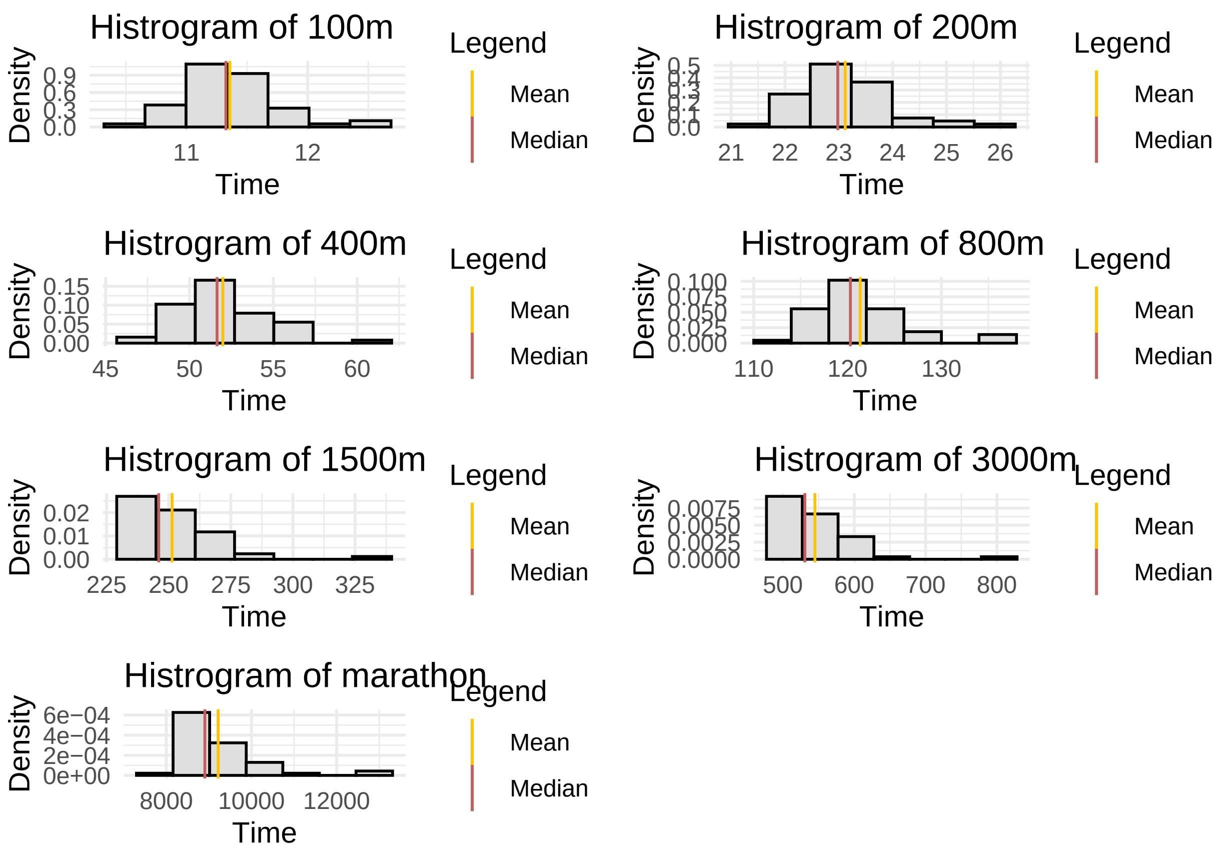
```
##        100m        200m        400m        800m       1500m       3000m 
##   0.3941012   0.9290255   2.5972019   5.2123825  16.3419012  48.9196132 
##    marathon 
## 986.3937051
```

We see that the mean exceeds the anticipated scaled times. This means that if the time for 100m is 11.3577778, one could assume that for 200m twice the time, 22.7155556 is needed. But due to human exhaustion we see that the means increase by more than that, in this case the mean for the 200m marathon is 23.1185185. For the median the effect is a bit weaker as it is not as heavily effected by outliers as the mean. Still the effect is visible. Also the deviations increase, here the described effect is even larger.

1.2 Illustrate the Variables

Task: Illustrate the variables with different graphs (explore what plotting possibilities R has). Make sure that the graphs look attractive (it is absolutely necessary to look at the labels, font sizes, point types). Are there any apparent extreme values? Do the variables seem normally distributed? Plot the best fitting (match the mean and standard deviation, i.e. method of moments) Gaussian density curve on the data's histogram. For the last part you may be interested in the `hist()` and `density()` functions.

Answer:



2 Relationships Between the Variables

2.1 Covariance and Correlation Matrices

Task: Compute the covariance and correlation matrices for the 7 variables. Is there any apparent structure in them? Save these matrices for future use.

Answer:

2.2 Scatterplots

Task: Generate and study the scatterplots between each pair of variables. Any extreme values?

Answer:

2.3 More Graphs

Task: Explore what other plotting possibilities R offers for multivariate data. Present other (at least two) graphs that you find interesting with respect to this data set.

Answer:

3 Examining for Extreme Values

3.1 Most Extreme Countries

Task: Look at the plots (esp. scatterplots) generated in the previous question. Which 3–4 countries appear most extreme? Why do you consider them extreme?

Answer:

3.2 Multivariate Residual

One approach to measuring "extremism" is to look at the distance (needs to be defined!) between an observation and the sample mean vector, i.e. we look how far one is from the average. Such a distance can be called an *multivariate residual* for the given observation.

Task: The most common residual is the Euclidean distance between the observation and sample mean vector, i.e.

$$d(\vec{x}, \bar{x}) = \sqrt{(\vec{x} - \bar{x})^T (\vec{x} - \bar{x})}.$$

This distance can be immediately generalized to the $L^r, r > 0$ distance as

$$d_{L^r}(\vec{x} - \bar{x}) = \left(\sum_{i=1}^{p} |\vec{x}_i - \bar{x}_i|^r \right)^{1/r},$$

where p is the dimension of the observation (here $p = 7$).

Compute the squared Euclidean distance (i.e. r = 2) of the observation from the sample mean for all 55 countries using R's matrix operations. First center the raw data by the means to get $\vec{x} - \bar{x}$ for each country. Then do a calculation with matrices that will result in a matrix that has on its diagonal the requested squared distance for each country. Copy this diagonal to a vector and report on the five most extreme countries. In this questions you **MAY NOT** use any loops.

Answer:

3.3 Squared Distance

Task: The different variables have different scales so it is possible that the distances can be dominated by some few variables. To avoid this we can use the squared distance

$$d^2_{\mathbf{V}}(\vec{x} - \bar{x}) = (\vec{x} - \bar{x})\boldsymbol{V}^{-1}(\vec{x} - \bar{x}),$$

where $\mathbf{V}$ is a diagonal matrix with variances of the appropriate variables on the diagonal. The effect, is that for each variable the squared distance is divided by its variance and we have a scaled independent distance. It is simple to compute this measure by standardizing the raw data with both means (centring) and standard deviations (scaling), and then compute the Euclidean distance for the normalized data. Carry out these computations and conclude which countries are the most extreme ones. How do your conclusions compare with the unnormalized ones?

Answer:

3.4 Mahalanobis Distance

Task: The most common statistical distance is the *Mahalanobis distance*

$$d_M^2(\vec{x} - \bar{x}) = (\vec{x} - \bar{x})^T \boldsymbol{C}^{-1}(\vec{x} - \bar{x}),$$

where **C** is the sample covariance matrix calculated from the data. With this measure we also use the relationships (covariances) between the variables (and not only the marginal variances as $d_V(\cdot, \cdot)$ does). Compute the Mahalanobis distance, which countries are most extreme now?

Answer:

3.5 Czekanowski's Diagram

Task: Compare the results in b)–d). Some of the countries are in the upper end with all the measures and perhaps they can be classified as extreme. Discuss this. But also notice the different measures give rather different results (how does Sweden behave?). Summarize this graphically. Produce Czekanowski's diagram using e.g. the `RMaCzek` package. In case of problems please describe them.

Answer:

4 Source Code

```r
library(gridExtra)
library(ggplot2)
knitr::opts_chunk$set(echo = TRUE)

track_times = read.table("data/T1-9.dat")
colnames(track_times) = c("country", "100m", "200m", "400m",
                          "800m", "1500m", "3000m", "marathon")
head(track_times)


track_times[,5:8] = track_times[,5:8] * 60
head(track_times)


track_times_mean = apply(track_times[,2:8], 2, mean)
track_times_median = apply(track_times[,2:8], 2, median)
track_times_sd = apply(track_times[,2:8], 2, sd)

track_times_mean
track_times_median
track_times_sd


p1 = ggplot() +
  geom_histogram(aes(x = track_times$`100m`, y=..density..),
                 color = "black", fill = "#dedede", bins = sqrt(nrow(track_times))) +
  geom_vline(aes(xintercept = track_times_mean[1], color="Mean")) +
  geom_vline(aes(xintercept = track_times_median[1], color ="Median")) +
  labs(title = "Histrogram of 100m",
       y = "Density",
       x = "Time", color = "Legend") +
  scale_color_manual(values = c("#FFC300", "#C25E5E")) +
  theme_minimal()

p2 = ggplot() +
  geom_histogram(aes(x = track_times$`200m`, y=..density..),
                 color = "black", fill = "#dedede", bins = sqrt(nrow(track_times))) +
  geom_vline(aes(xintercept = track_times_mean[2], color = "Mean")) +
  geom_vline(aes(xintercept = track_times_median[2], color = "Median")) +
  labs(title = "Histrogram of 200m",
       y = "Density",
       x = "Time", color = "Legend") +
  scale_color_manual(values = c("#FFC300", "#C25E5E")) +
  theme_minimal()

p3 = ggplot() +
  geom_histogram(aes(x = track_times$`400m`, y=..density..),
                 color = "black", fill = "#dedede", bins = sqrt(nrow(track_times))) +
  geom_vline(aes(xintercept = track_times_mean[3], color = "Mean")) +
  geom_vline(aes(xintercept = track_times_median[3], color = "Median")) +
  labs(title = "Histrogram of 400m",
       y = "Density",
       x = "Time", color = "Legend") +
  scale_color_manual(values = c("#FFC300", "#C25E5E")) +
  theme_minimal()

p4 = ggplot() +
  geom_histogram(aes(x = track_times$`800m`, y=..density..),
                 color = "black", fill = "#dedede", bins = sqrt(nrow(track_times))) +
  geom_vline(aes(xintercept = track_times_mean[4],  color = "Mean")) +
  geom_vline(aes(xintercept = track_times_median[4],  color = "Median")) +
  labs(title = "Histrogram of 800m",
       y = "Density",
       x = "Time", color = "Legend") +
  scale_color_manual(values = c("#FFC300", "#C25E5E")) +
  theme_minimal()

p5 = ggplot() +
  geom_histogram(aes(x = track_times$`1500m`, y=..density..),
                 color = "black", fill = "#dedede", bins = sqrt(nrow(track_times))) +
  geom_vline(aes(xintercept = track_times_mean[5], color = "Mean")) +
  geom_vline(aes(xintercept = track_times_median[5], color = "Median")) +
  labs(title = "Histrogram of 1500m",
       y = "Density",
       x = "Time", color = "Legend") +
  scale_color_manual(values = c("#FFC300", "#C25E5E")) +
  theme_minimal()

p6 = ggplot() +
  geom_histogram(aes(x = track_times$`3000m`, y=..density..),
                 color = "black", fill = "#dedede", bins = sqrt(nrow(track_times))) +
  geom_vline(aes(xintercept = track_times_mean[6], color = "Mean")) +
  geom_vline(aes(xintercept = track_times_median[6], color = "Median")) +
  labs(title = "Histrogram of 3000m",
       y = "Density",
       x = "Time", color = "Legend") +
  scale_color_manual(values = c("#FFC300", "#C25E5E")) +
  theme_minimal()

p7 = ggplot() +
  geom_histogram(aes(x = track_times$`marathon`, y=..density..),
                 color = "black", fill = "#dedede", bins = sqrt(nrow(track_times))) +
  geom_vline(aes(xintercept = track_times_mean[7], color = "Mean")) +
  geom_vline(aes(xintercept = track_times_median[7], color = "Median")) +
  labs(title = "Histrogram of marathon",
       y = "Density",
       x = "Time", color = "Legend") +
  scale_color_manual(values = c("#FFC300", "#C25E5E")) +
  theme_minimal()

grid.arrange(p1, p2, p3, p4, p5, p6, p7, ncol = 2)
```