Statistical Analysis of Electrocardiogram Data

Statistical Analysis of Electrocardiogram Data

By

Zhiyong Tang, B.Sc.

A Thesis

Submitted to the School of Graduate Studies

In Partial Fulfillment of the Requirements

for the Degree

Master of Science

McMaster University

MASTER OF SCIENCE (2009) McMaster University

(Statistics) Hamilton, Ontario

TITLE: Statistical Analysis of Electrocardiogram Data

AUTHOR: Zhiyong Tang

SUPERVISOR: Dr. Román Viveros-Aguilera

NUMBER OF PAGES: vi, 86

Abstract

In this thesis we focus on statistical analysis of electrocardiogram data. These data record the electrical activity of the heart muscle. The data used in this thesis were provided by Dr. Raimond Wong from Hamilton Regional Cancer Centre (HRCC). The number of independent cases is small (6 cases), but each electrocardiogram contains over 400000 plotting points. Three electrocardiograms came from cancer patients while the other 3 came from healthy volunteers.

We conduct statistical analysis in two stages: extraction of feature vectors and clustering analysis of feature vectors. During the first stage, we define 7 statistics that capture important features of the electrocardiogram data. Then these 7 features are separately used in a univariate way to classify the electrocardiogram data into two groups as patients and volunteers. Results show that some of the features can separate the electrocardiogram data well, but others can not do the job well.

During the stage of clustering analysis using the 7 features in a multivariate way, we employ three methods of clustering analysis: hierarchical clustering analysis, K-means clustering analysis, and Andrews plot clustering analysis. Results show that hierarchical clustering analysis and K-means clustering analysis misclassify one of the subjects. Andrews plot clustering analysis however successfully classify all the subjects. The first two methods are more objective while the latter requires more judgement. Note that the limited number of independent cases available does not support general conclusions, but our study suggest some potential for the methods discussed.

Acknowledgments

I would like to express my deep appreciation to my supervisor, Dr. Román Viveros-Aguilera, for his inspiring direction and generous support throughout the entire process of my thesis. Dr. Román Viveros-Aguilera is patient and understanding, and I am very lucky to have his great guidance.

I thank Dr. Peter Macdonald and Dr. Rong Zhu for their careful reading of my thesis and for their insightful suggestions which led to many improvements in the final version of my thesis.

I am also grateful to my professors, staff and my fellow graduate students in the Department of Mathematics and Statistics of McMaster University.

Extreme thanks to my parents and my sister for their encouragement and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction to Electrocardiogram Data

1.1 Introducing Basics of the Working of the Heart

The heart has two kinds of principal cells: working cells and specialized neural-like conductive cells. The muscle or myocardium of the atria and ventricles are the working cells. Specialized neural-like conductive cells include the Sinuatrial (SA) node, the Atrioventricular (AV) node, the Bundle of His, and the Purkinje fibers (Becker, 2006), which are shown in Figure 1.1.

The blood output of the heart per minute is the paramount cardiovascular event required to sustain blood flow throughout the whole body. In addition to blood volume and contractile strength, the muscle cells of the heart are linked very closely to one another, so that the electrical impulses can easily spread from one cell to the next. Cer-

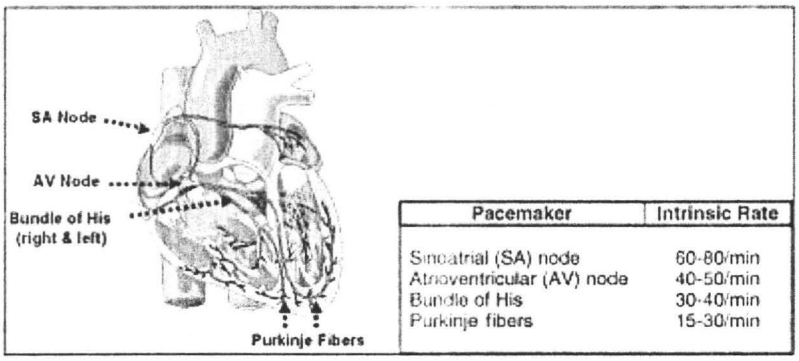


Figure 1.1: Specialized neural-like conductive heart tissues and their approximate rates (Becker, 2006).

tain groups of specialized neural-like conductive cells rapidly transmit electrical activity through the heart. The electrical activity of the heart muscle can be recorded from the body surface, monitored by a device called electrocardiogram (Khorovets, 2000). Electrocardiogram monitoring is regarded as a standard of care during general anesthesia and is strongly encouraged when providing deep sedation.

The electrical activity of the heart muscle comes from the process of "depolarization" of the heart muscle cells. The inside of the cardiac muscle cells is negatively charged with respect to the outside in its resting state, which is called in "polarized" status. When there is a greater concentration of certain charged ions on one side of the cell membrane as compared with the other side, the cardiac muscle cells are charged. For example, the concentration of potassium ions is much higher inside the cells while the concentration of sodium ions is much higher outside. These ions will move in response to

stimuli, particularly a rapid inward movement of sodium, and then a rapid loss of internal negative potential, which generates electricity. The opposite process of the heart muscle cells is called "repolarization". The processes of depolarization and repolarization are shown in Figure 1.2 with more details.

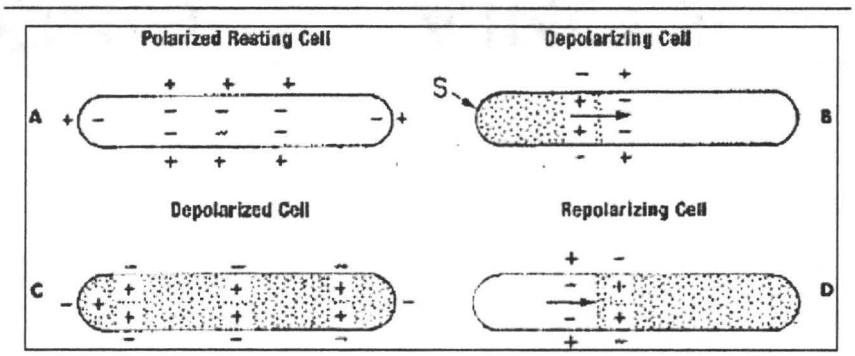


Figure 1.2: The process of depolarization and repolarization (Becker, 2006).

In Figure 1.2, the A step means that the resting cell membrane is charged positively on the outside and negatively on the inside. The B step tells that positive ions enter the cell, reversing this polarity following a stimulus S. The C step tells this process continues until the entire cell is depolarized. And the D step means that ions are returned to their normal location and the cell repolarizes to its normal resting potential.

1.2 Recording Electrocardiogram Data

The first crude electrocardiogram was introduced by a Dutch physiologist, Willem Einthoven in 1901. The electrocardiogram records the electrical activity of the heart muscle by 3 electrode arrangements, which are known as the primary limb leads I, II,

and III. Figure 1.3 gives more details.

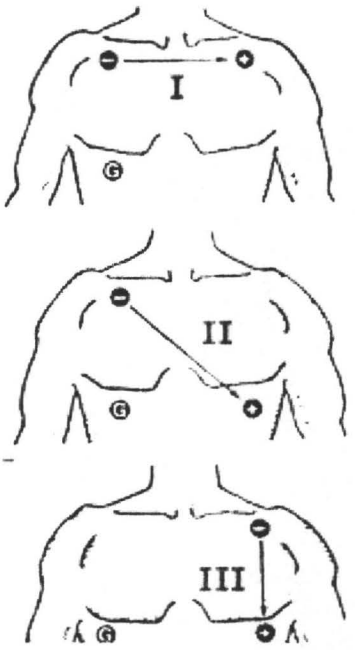


Figure 1.3: Standard limb leads I, II, and III (Becker, 2006).

In Figure 1.3, G means that electrode lead connects to the ground. Most often, lead II is selected as the important data source in research analysis because it generally records the largest electronic waves of the heart muscle cells.

When current electricity flow passes into the positive end of the bipolar (2-sided) electrode, it causes a positive deflection, which corresponds to an upward movement of the pen on the electrocardiogram paper. When current electricity flow passes away from the positive pole of the bipolar electrode, it causes a negative deflection and a downward movement of the pen on the electrocardiogram paper instead. A typical electrocardiogram is shown in Figure 1.4.

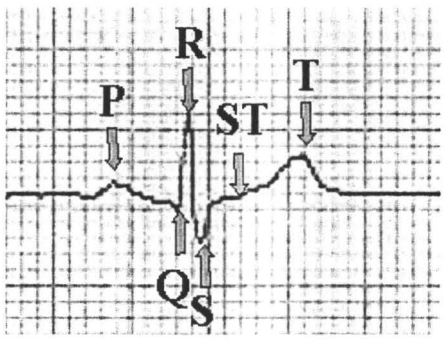
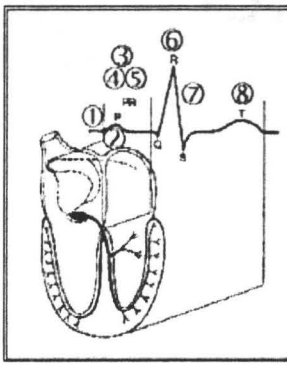


Figure 1.4: A standard electrocardiogram record (Khorovets, 2000).

Figure 1.5 gives more details about the meaning of the segment of waves in an electrocardiogram in the processes of depolarization and repolarization of the heart cells.



	Physiologic Event	ECG Evidence
1.	SA node initiates impulse	Not visible
2.	Depolarization of atrial muscle	P wave
3.	Atrial contraction	Not visible
4.	Depolarization of AV node & Common Bundle	Not visible
5.	Repolarization of atrial muscle	Not visible
6.	Depolarization of ventricular muscle	QRS complex
7.	Contraction of ventricular muscle	Not visible
8.	Repolarization of ventricular muscle	T wave

Figure 1.5: Summary of events of a cardiac cycle (Becker, 2006).

From Figure 1.5, we can see that of the 8 physiologic events listed for a cardiac cycle, only 3 are actually observed on the electrocardiogram. The observable events are depicted in Figure 1.4.

Chapter 2

Statistical Modeling of Electrocardiogram Data

The main motivation for this thesis is the work of Dr. Raimond Wong and his team from the Hamilton Regional Cancer Center (HRCC). In fact, all the electrocardiogram data used in this thesis were kindly provided by Dr. Wong. One of the key studies of Dr. Wong and his team carried out at the HRRC focused on evaluating heart rate variability and its relationship with cancer related fatigue syndrome in gut, breast, and prostate cancer patients (Wong, 2004). The cancer patients in radiation therapy may feel tired at various times during the therapy, so the whole objective of this research study was to find a better way to assess how tired the cancer patients are, by taking and studying the electrocardiograms of the patients. In the research study, the electrocardiograms of healthy volunteers were also taken to have a basis for comparison with those of cancer patients. During recording of the electrocardiograms, the electrodes have been attached

to the wires, and they are placed on the subject in the way referred to the Lead II position in Chapter 1. Table 2.1 shows more details on the positions of the wires.

Table 2.1: Positions of the wires in Lead II (Wong, 2004).

Wire	**Location**
Positive	Left side of the stomach
Negative	Under right collar bone
Ground	Under left collar bone

In the research study, the electrocardiogram should be taken at a minimum sampling rate of 500Hz for at least 5 minutes for the subject as recommended by the Task Force of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology. So the sampling rate used for all electrocardiograms in Wong's research study is 1000Hz with a minimum measuring time of 7.5 minutes, and the electrocardiograms were taken with the subjects in a stationary position.

So our electrocardiogram data (recorded by lead II) are from 3 cancer patients and 3 healthy volunteers, and the data were taken on a sitting position for every patient and healthy volunteer. The sampling frequency is $f = 1000$Hz. The total test time is almost 7.5 minutes for every patient and healthy volunteer.

The main objective of this thesis are twofold:

- To find out differences between the electrocardiogram data of cancer patients and

those for healthy individuals as far as statistical measures is concerned.

- To develop statistical methods to classify cancer patients and healthy individuals based on their electrocardiogram data.

The main idea is to establish if the electrocardiogram data can be used to predict whether an individual is cancer-free or not. As far as we know, there is no such research study focusing on this topic of the electrocardiogram data.

2.1 Original Electrocardiogram Data

The initial data sets that Dr. Wong produced consisted of the electrocardiograms of 3 cancer patients and 3 healthy volunteers. Table 2.2 gives some details of the electrocardiogram data for all the patients and healthy volunteers.

Typical of electrocardiogram data, the number of points per electrocardiogram is very large. Figures 2.1-2.2 display separately for the first 10000 data points of the original electrocardiograms for Patient 1, Patient 2, Patient 3 and Volunteer 1, Volunteer 2, Volunteer 3 in the same magnitude range. We only plot 10000 points so that the features can be seen clearly.

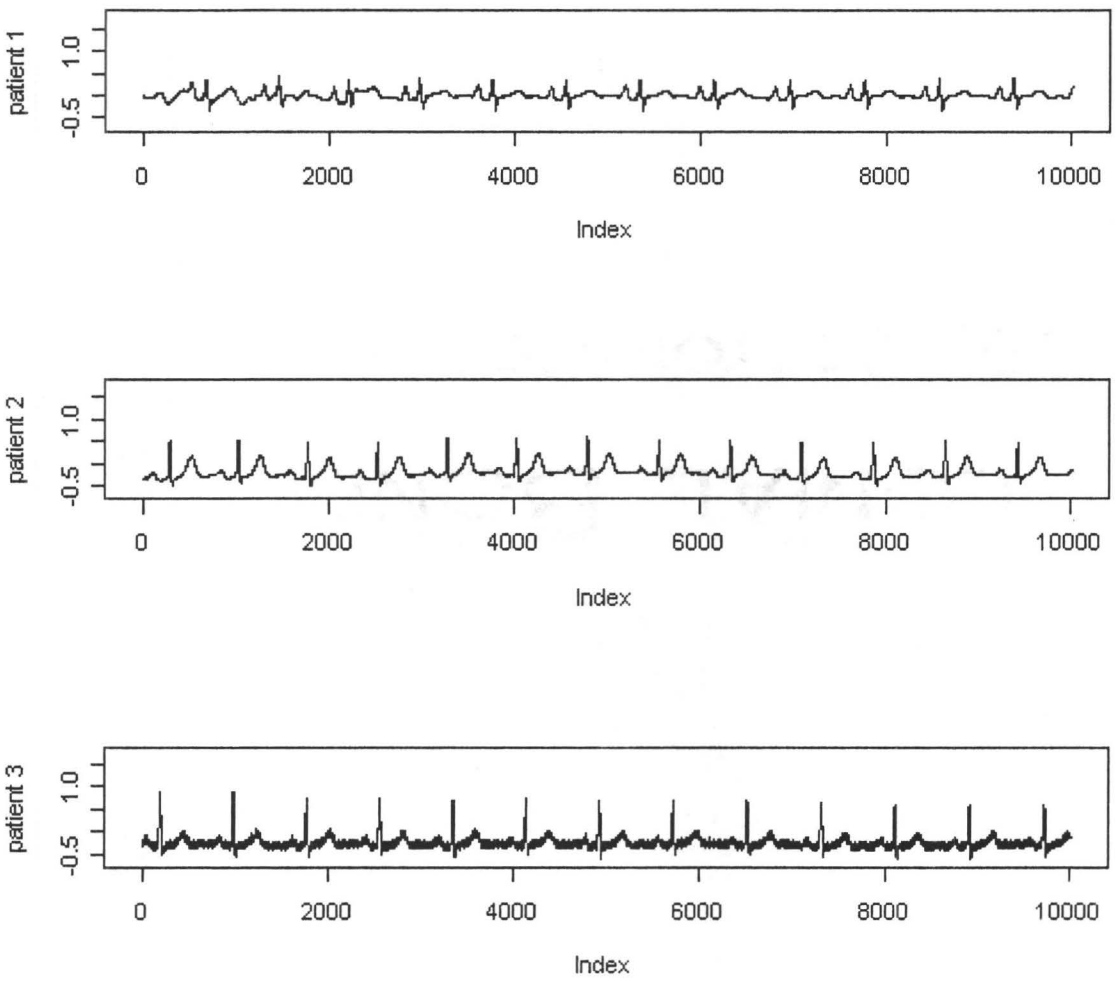


Figure 2.1: Original electrocardiogram data of Patients.

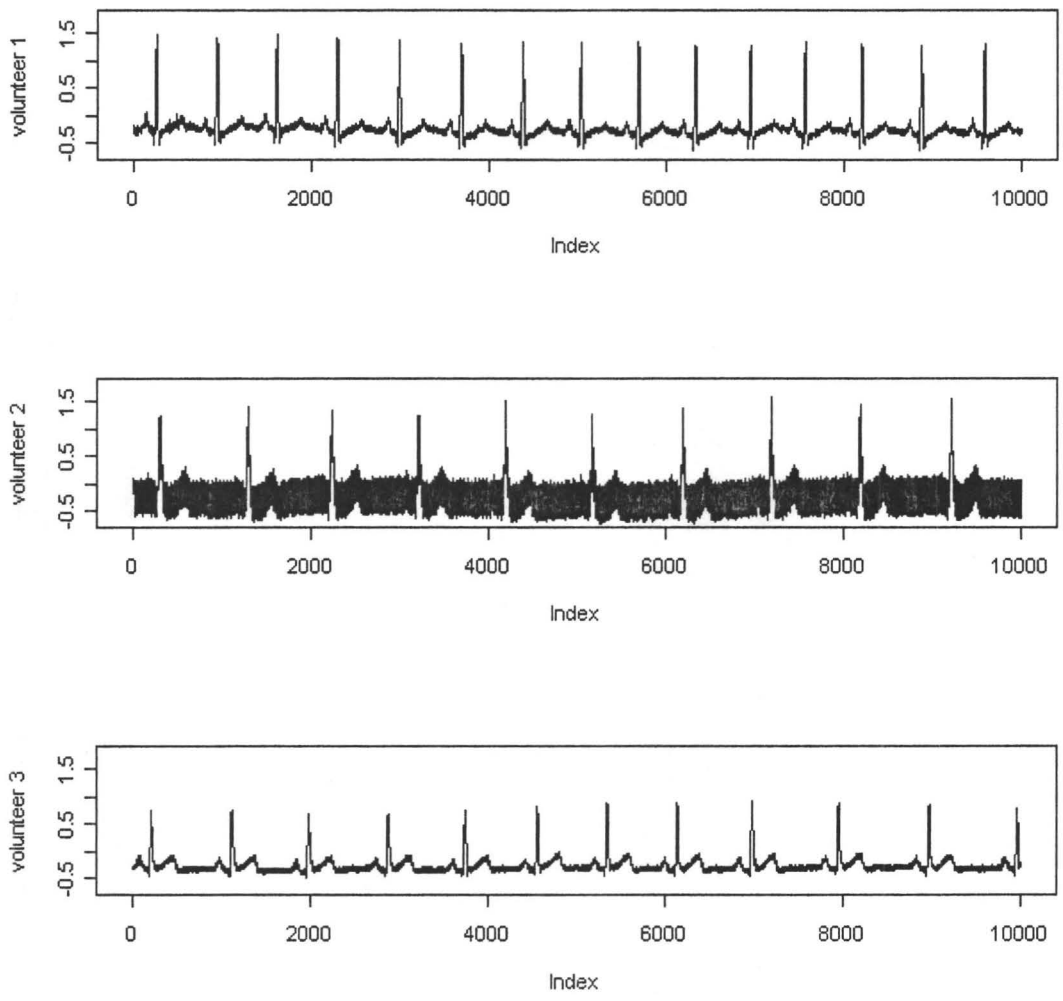


Figure 2.2: Original electrocardiogram data of Volunteers.

Table 2.2: Total data time and data points of the electrocardiogram data.

	Total Data Time (Min)	Total Data Points
Patient 1	7.544	452618
Patient 2	7.647	458810
Patient 3	7.575	454510
Volunteer 1	7.544	452618
Volunteer 2	7.532	451930
Volunteer 3	7.613	456746

2.2 Segmenting Electrocardiogram Data

From a statistical point of view, every electrocardiogram is a curve, i.e., a mathematical function. Thus we are dealing with functional data, a type of data that has been the focus of intense activity in the last few years (Ramsay and Silverman, 2002). Ramsay and Silverman (2005) had also given an excellent overview in the analysis of functional data. The curves are continuous but naturally observations (measurements) are only possible at discrete (time) points. For instance, the function for Patient 1 was observed at 452618 points (Table 2.2). As we can notice from Figure 2.1, high and low peak points of varying height and diverse speeds of moving up and down are quite apparent. A natural approach to understand and analyze this kind of functional data is to start by extracting statistical shape features from the curves to form a feature vector. In this section we describe seven shape features that are quite noticeable in the electrocardiogram data.

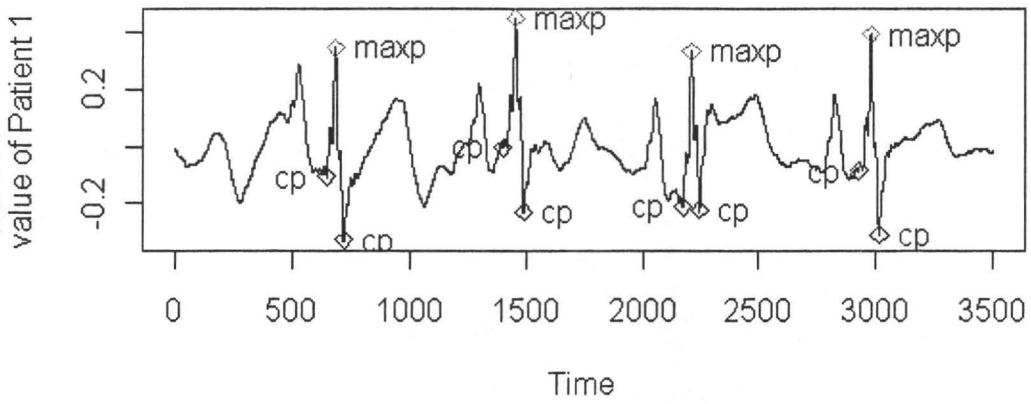


Figure 2.3: Segment of electrocardiogram data of Patient 1.

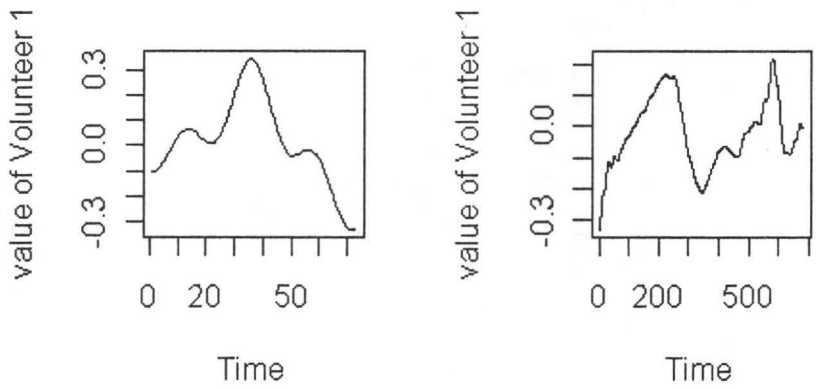


Figure 2.4: Higher peak (left) and lower peak (right) of electrocardiogram data of Patient 1.

Figure 2.3 gives an example of breaking the electrocardiogram data of Patient 1 into higher peaks (corresponding to QRS wave in Chapter 1) and lower peaks (corresponding to P wave and T wave in Chapter 1). We record the cut point ("cp" in figure) time index into a vector $\mathbf{P} = (p_1, p_2, p_3, p_4, p_5, p_6, \cdots)^T$ and also record the maximum point ("maxp" in figure) time index into a vector $\mathbf{M} = (m_1, m_2, m_3, m_4, \cdots)^T$. The higher peaks are between p_n and p_{n+1} : (p_n, p_{n+1}), when n is an odd number, and the lower peaks are between p_n and p_{n+1} : (p_n, p_{n+1}), when n is an even number.

Figure 2.4 gives us more details for one example of the higher peak (left) and lower peak (right) for the electrocardiogram data of Patient 1.

Figures 2.5-2.6 give one example of breaking the electrocardiogram data of Volunteer 1 into higher peaks and lower peaks, and the details for the higher peak (left) and lower peak (right) for Volunteer 1.

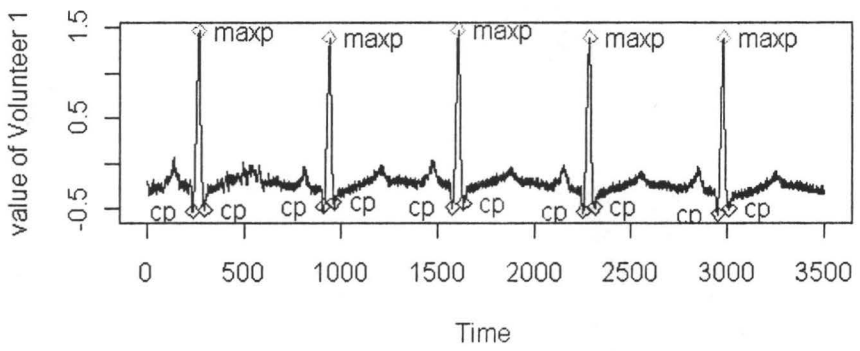


Figure 2.5: Segment of electrocardiogram data of Volunteer 1.

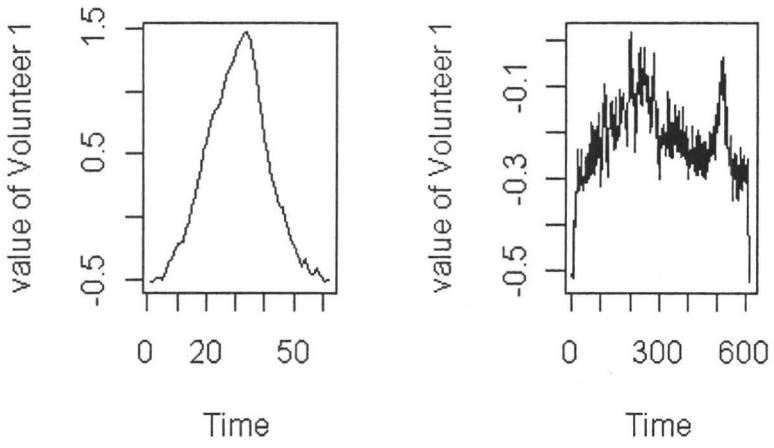


Figure 2.6: Higher peak (left) and lower peak (right) of electrocardiogram data of Volunteer 1.

We apply the same method to all other electrocardiograms of patients and volunteers.

2.3 Statistical Modeling for Electrocardiogram Data

According to Section 2.2, we decompose the electrocardiogram data of all patients and volunteers into two processes: higher peaks and lower peaks. In addition to these most noticeable features, we can have more other features which capture the most important shape features of an electrocardiogram.

2.3.1 Univariate Shape Features for Electrocardiogram Data

Without loss of generality, we continue to use the electrocardiogram data of Patient 1 as an example. Let $\mathbf{Y} = (y_1, y_2, y_3, y_4, \cdots)^T$ be the data value vector for Patient 1, then the following seven important statistical features can be calculated:

1. Maximum value of higher peak (p_{2k-1}, p_{2k}): $p1_{m_k}$, $k = 1, 2, 3, \cdots$

2. Range of higher peak (p_{2k-1}, p_{2k}): $p1_{RH_k} = y_{m_k} - \min(y_{p_{2k-1}}, \cdots, y_{p_{2k}})$, $k = 1, 2, 3, \cdots$

3. Maximum value of lower peak (p_{2k}, p_{2k+1}): $p1_{mL_k} = \max(y_{p_{2k}}, \cdots, y_{p_{2k+1}})$, $k = 1, 2, 3, \cdots$

4. Range of lower peak (p_{2k}, p_{2k+1}): $p1_{RL_k} = y_{ml_k} - \min(y_{p_{2k}}, \cdots, y_{p_{2k+1}})$, $k = 1, 2, 3, \cdots$

5. Width of higher peak (p_{2k-1}, p_{2k}): $p1_{WH_k} = (p_{2k} - p_{2k-1})/f$, $f = 1000hz$, $k = 1, 2, 3, \cdots$

6. Width of lower peak (p_{2k}, p_{2k+1}): $p1_{WL_k} = (p_{2k+1} - p_{2k})/f$, $f = 1000hz$, $k = 1, 2, 3, \cdots$

7. Average value of lower peak (p_{2k}, p_{2k+1}): $p1_{AVL_k} = (y_{p_{2k}} + \cdots + y_{p_{2k+1}})/(p_{2k+1} - p_{2k} + 1)$, $k = 1, 2, 3, \cdots$

So for the kth segment (p_{2k-1}, p_{2k+1}) in the electrocardiogram data of Patient 1, we

have an corresponding feature vector of seven important statistics:

$$\mathbf{P1}_{ht_k} = (p1_{m_k}, p1_{RH_k}, p1_{mL_k}, p1_{RL_k}, p1_{WH_k}, p1_{WL_k}, p1_{AVL_k})^T,\ k = 1, 2, 3, \cdots.$$

Then the group set of feature vectors: $\{\mathbf{P1}_{ht_k}, k = 1, 2, 3, \cdots\}$ can be used to represent the electrocardiogram data of Patient 1 for further study. We apply the same method to all the other electrocardiogram data of patients and volunteers, then we have:

$\{\mathbf{P2}_{ht_k}, k = 1, 2, 3, \cdots\}$ $(\mathbf{P2}_{ht_k} = (p2_{m_k}, p2_{RH_k}, p2_{mL_k}, p2_{RL_k}, p2_{WH_k}, p2_{WL_k}, p2_{AVL_k})^T)$ for Patient 2,

$\{\mathbf{P3}_{ht_k}, k = 1, 2, 3, \cdots\}$ $(\mathbf{P3}_{ht_k} = (p3_{m_k}, p3_{RH_k}, p3_{mL_k}, p3_{RL_k}, p3_{WH_k}, p3_{WL_k}, p3_{AVL_k})^T)$ for Patient 3,

$\{\mathbf{V1}_{ht_k}, k = 1, 2, 3, \cdots\}$ $(\mathbf{V1}_{ht_k} = (v1_{m_k}, v1_{RH_k}, v1_{mL_k}, v1_{RL_k}, v1_{WH_k}, v1_{WL_k}, v1_{AVL_k})^T)$ for Volunteer 1,

$\{\mathbf{V2}_{ht_k}, k = 1, 2, 3, \cdots\}$ $(\mathbf{V2}_{ht_k} = (v2_{m_k}, v2_{RH_k}, v2_{mL_k}, v2_{RL_k}, v2_{WH_k}, v2_{WL_k}, v2_{AVL_k})^T)$ for Volunteer 2,

$\{\mathbf{V3}_{ht_k}, k = 1, 2, 3, \cdots\}$ $(\mathbf{V3}_{ht_k} = (v3_{m_k}, v3_{RH_k}, v3_{mL_k}, v3_{RL_k}, v3_{WH_k}, v3_{WL_k}, v3_{AVL_k})^T)$ for Volunteer 3.

The following figures are the boxplots and density plots of the seven statistical features for the electrocardiogram data, separately.

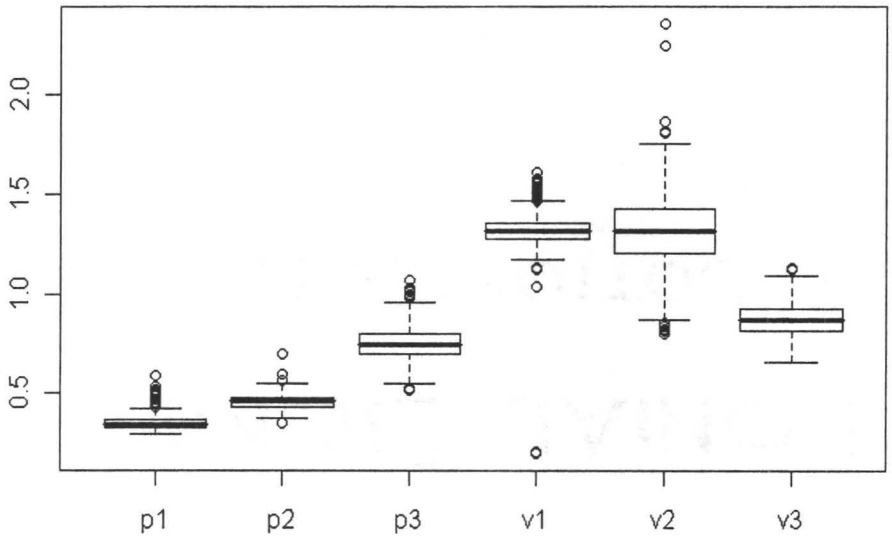


Figure 2.7: Boxplot of the maximum value of higher peak.

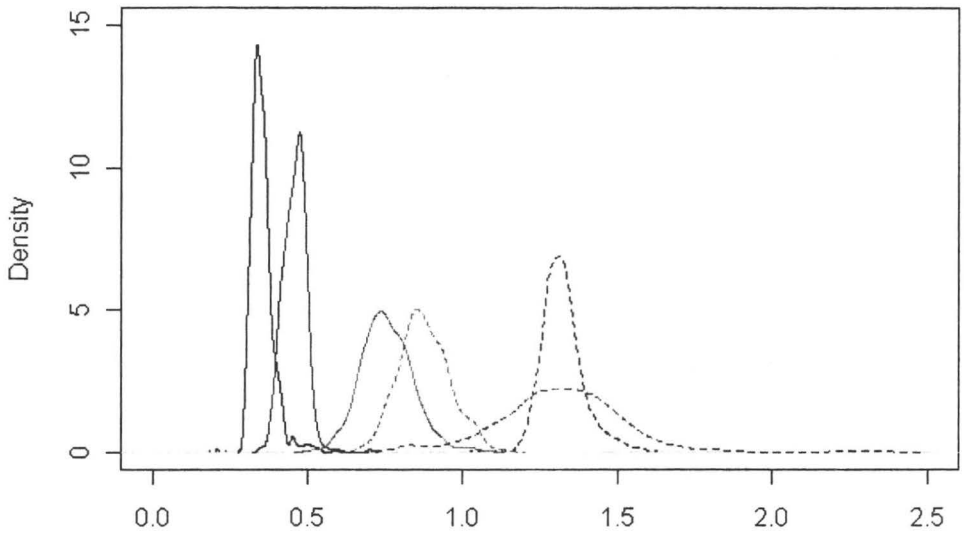


Figure 2.8: Density of the maximum value of higher peak (Solid lines: Patients, Dash Lines: Volunteers).

Figures 2.7-2.8 are the boxplots and pdfs of the maximum of higher peak of the electrocardiogram data: $\{\{p1_{m_k}\}, \{p2_{m_k}\}, \{p3_{m_k}\}, \{v1_{m_k}\}, \{v2_{m_k}\}, \{v3_{m_k}\}\}$, $k = 1, 2, 3, \cdots$ for V1, V2, V3, P1, P2, P3. From Figure 2.7, the center of the boxplots of V1, V2, and V3 are greater than 0.8 (between 1.2 and 1.4 for V1, between 1.0 and 1.5 for V2, between 0.8 and 0.9 for V3), but the center of P1, P2 and P3 are less than 0.8 (between 0.3 and 0.4 for P1, between 0.4 and 0.5 for V2, between 0.7 and 0.8 for V3). Figure 2.8 (Solid lines: left for P1, middle for P2, right for P3. Dash lines: high for V1, middle for V3, low for V2) shows the same property for the pdfs of the maximum of higher peak,

and the pdfs of patients are more compact (less variance) than the pdfs of volunteers.

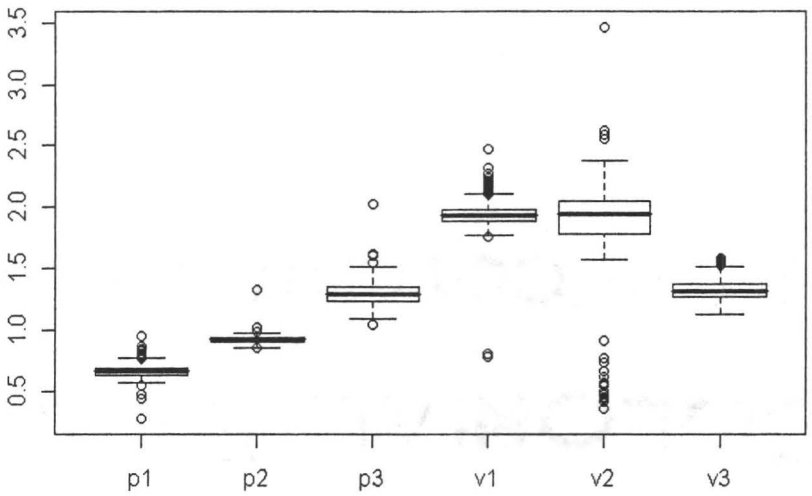


Figure 2.9: Boxplot of the range value of higher peak.

Figures 2.9-2.10 are the boxplots and pdfs of the range value of higher peak of the electrocardiogram data: $\{\{p1_{RH_k}\}, \{p2_{RH_k}\}, \{p3_{RH_k}\}, \{v1_{RH_k}\}, \{v2_{RH_k}\}, \{v3_{RH_k}\}\}$, $k = 1, 2, 3, \cdots$ for V1, V2, V3, P1, P2, P3, respectively. From the boxplots we can see that the center of the boxplots of V1 and V2 are around 2.0, and the center of V3 is around 1.5, but the center of P1, P2 and P3 are less than 1.5 (between 0.6 and 0.7 for P1, between 0.9 and 1.0 for V2, between 1.0 and 1.5 for V3). Figure 2.10 (Solid lines: left for P1, middle for P2, right for P3. Dash lines: high for V1, middle for V3, low for V2) shows the same property from the pdfs of the range value of higher peak for V1, V2, V3, P1, P2, P3. The pdfs of P1, P2, P3 are more compact (less variance) than the

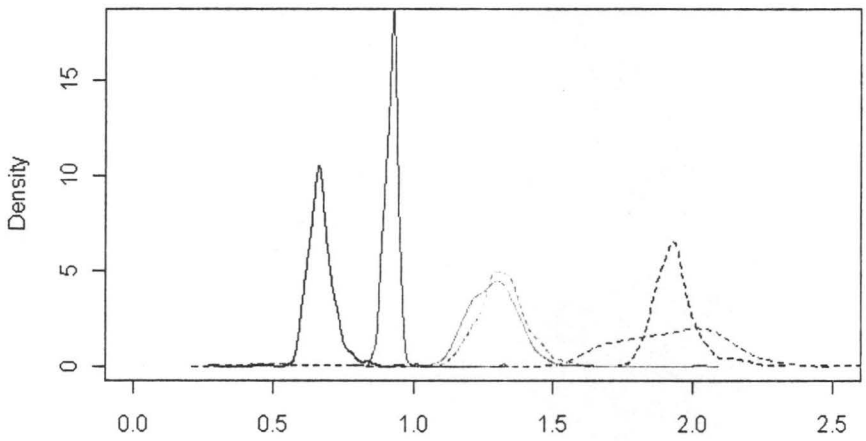


Figure 2.10: Density of the range value of higher peak (Solid lines: Patients, Dash Lines: Volunteers).

pdfs of V1, V2, V3.

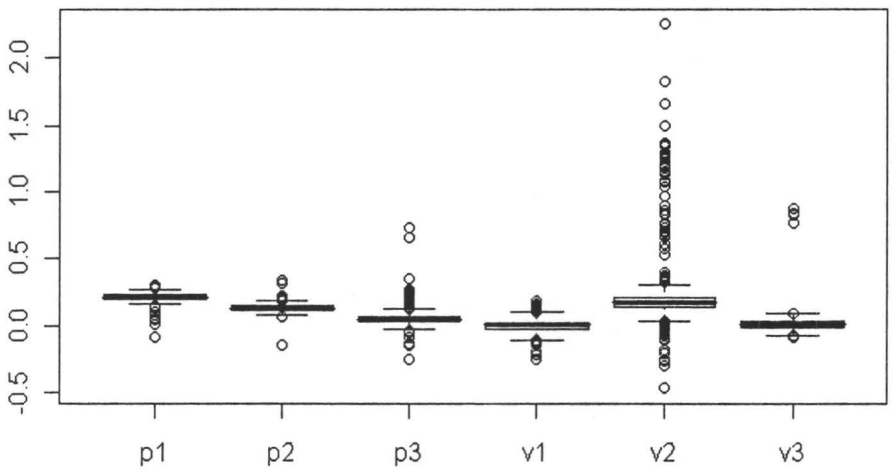


Figure 2.11: Boxplot of the maximum value of lower peak.

Figures 2.11-2.12 are the boxplots and pdfs of the maximum value of lower peak of the electrocardiogram data: $\{\{p1_{mL_k}\}, \{p2_{mL_k}\}, \{p3_{mL_k}\}, \{v1_{mL_k}\}, \{v2_{mL_k}\}, \{v3_{mL_k}\}\}$, $k = 1, 2, 3, \cdots$ for V1, V2, V3, P1, P2, P3, respectively. From the boxplots we can see that the centers of the boxplots of V1, V3 and P3 are around 0, but the center of P2 are around 0.2, and the centers of P1 and V2 are around 0.15. Figure 2.12 (Solid lines: left for P3, middle for P2, right for P1. Dash lines: high for V3, middle for V1, low for V2) shows the same property from the pdfs of the maximum value of lower peak for V1, V2, V3, P1, P2, P3.

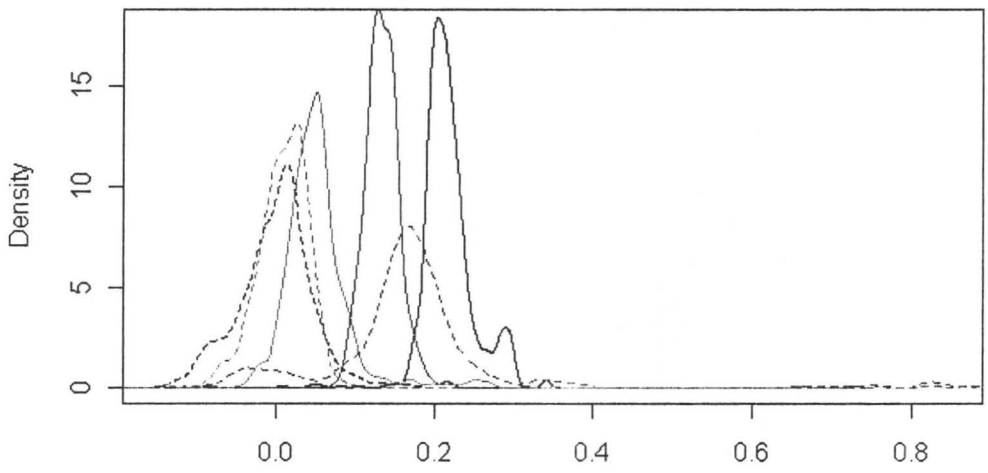


Figure 2.12: Density of the maximum value of lower peak (Solid lines: Patients, Dash Lines: Volunteers).

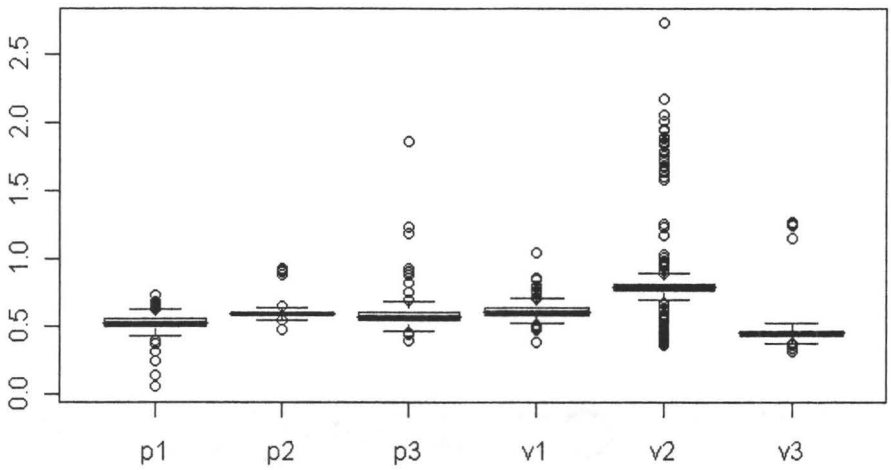


Figure 2.13: Boxplot of the range value of lower peak.

Figures 2.13-2.14 are the boxplots and pdfs of the range value of lower peak of the electrocardiogram data: $\{\{p1_{RL_k}\}, \{p2_{RL_k}\}, \{p3_{RL_k}\}, \{v1_{RL_k}\}, \{v2_{RL_k}\}, \{v3_{RL_k}\}\}$, $k = 1, 2, 3, \cdots$ for V1, V2, V3, P1, P2, P3, respectively. From the boxplots we can see that the centers of the boxplot of V2 is around 0.8, and the center of all other boxplots of V1, V3, P1, P2, P3 are around 0.5. Figure 2.14 (Dash lines: left for V3, middle for V1, right for V2. Solid lines: high for P2, middle for P1, low for P3) shows the same property from the pdfs of the range value of lower peak for V1, V2, V3, P1, P2, P3.

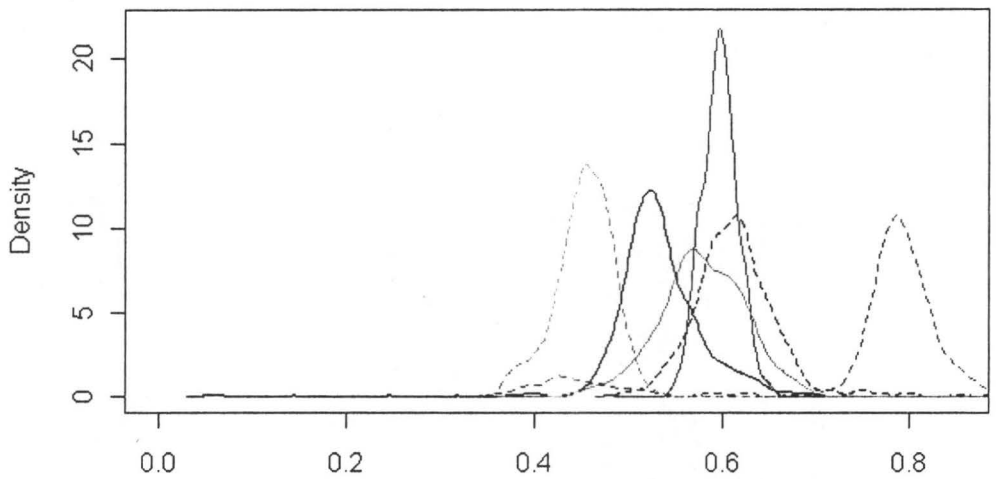


Figure 2.14: Density of the range value of lower peak (Solid lines: Patients, Dash Lines: Volunteers).

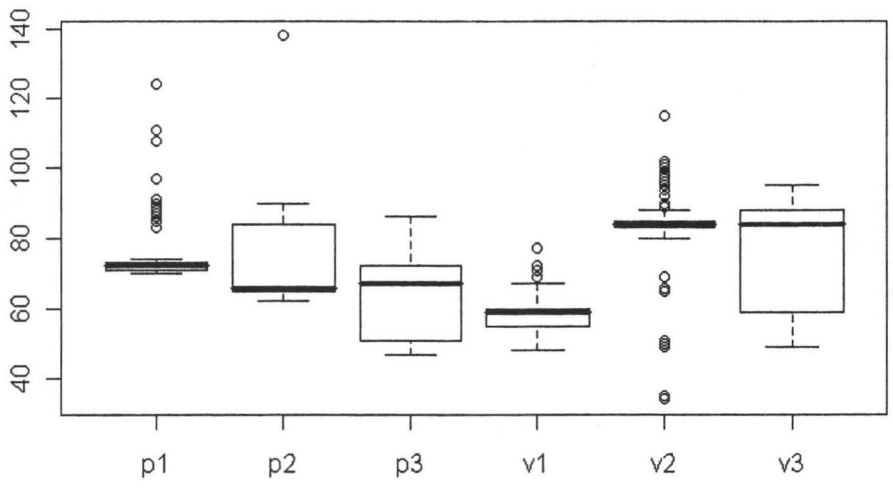


Figure 2.15: Boxplot of the width of higher peak.

Figures 2.15-2.16 are the boxplots and pdfs of the width of higer peak of the electrocardiogram data: $\{\{p1_{WH_k}\}, \{p2_{WH_k}\}, \{p3_{WH_k}\}, \{v1_{WH_k}\}, \{v2_{WH_k}\}, \{v3_{WH_k}\}\}$, $k = 1, 2, 3, \cdots$ for V1, V2, V3, P1, P2, P3, respectively. From the boxplots we can see that the centers of the boxplots of V2, and V3 are around 80, the center of V1 is around 60. But the center of P1, P2 and P3 are around 70. In Figure 2.16 (Dash lines: high for V2, middle for V1, low for V3. Solid lines: high for P1, middle for P2, low for P3), we can also see the same pattern from the pdfs of the width of higher peak for V1, V2, V3, P1, P2, P3.

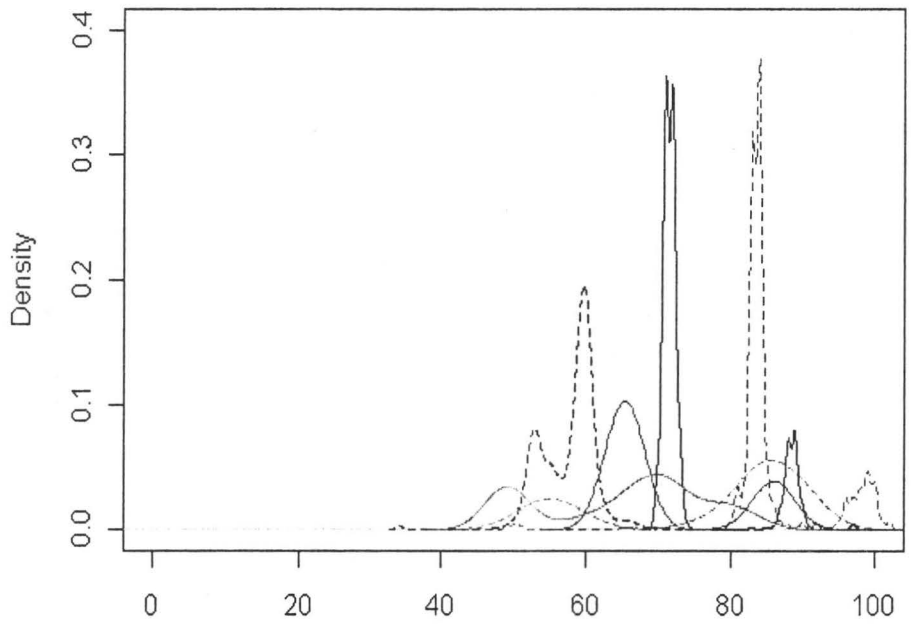


Figure 2.16: Density of the width of higher peak (Solid lines: Patients, Dash Lines: Volunteers).

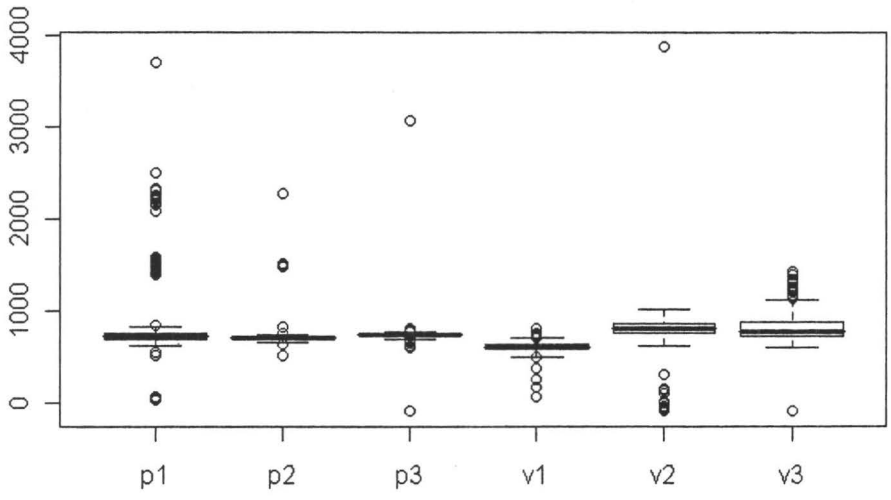


Figure 2.17: Boxplot of the width of lower peak.

Figures 2.17-2.18 are the boxplots and pdfs of the width of lower peak of the electrocardiogram data: $\{\{p1_{WL_k}\}, \{p2_{WL_k}\}, \{p3_{WL_k}\}, \{v1_{WL_k}\}, \{v2_{WL_k}\}, \{v3_{WL_k}\}\}$, $k = 1, 2, 3, \cdots$ for V1, V2, V3, P1, P2, P3, respectively. From the boxplots we can see that the centers of the boxplots of V2, V3, P1, P2, P3 are all around 750, but the center of the boxplot of V1 is around 650. In Figure 2.18 (Dash lines: high for V1, middle for V2, low for V3. Solid lines: high for P2, middle for P3, low for P1), we can also see the same pattern from the pdfs of the width of lower peak for V1, V2, V3, P1, P2, P3.

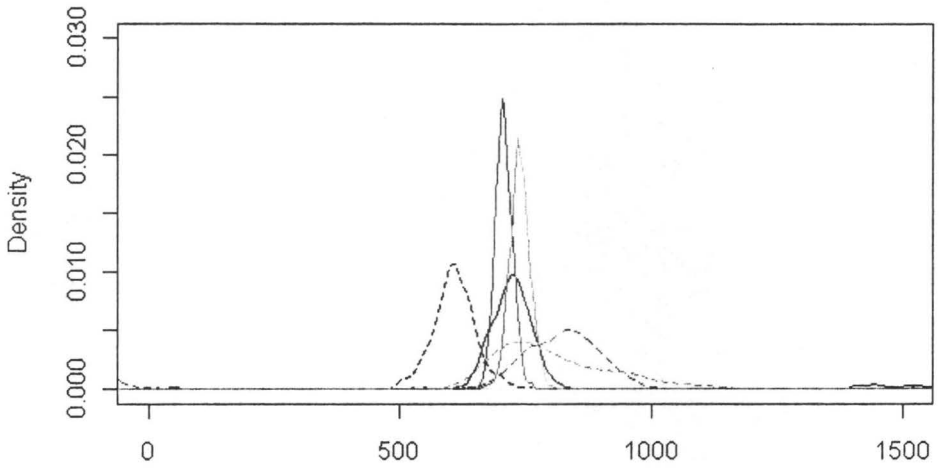


Figure 2.18: Density of the width of lower peak (Solid lines: Patients, Dash Lines: Volunteers).

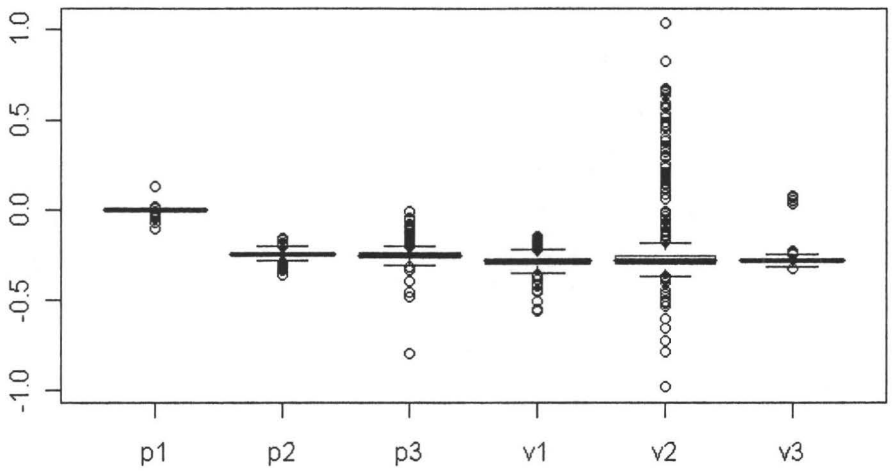


Figure 2.19: Boxplot of the mean value of lower peak.

Figures 2.19-2.20 are the boxplots and pdfs of the mean value of lower peak of the electrocardiogram data: $\{\{p1_{AVL_k}\}, \{p2_{AVL_k}\}, \{p3_{AVL_k}\}, \{v1_{AVL_k}\}, \{v2_{AVL_k}\}, \{v3_{AVL_k}\}\}$, $k = 1, 2, 3, \cdots$ for V1, V2, V3, P1, P2, P3, respectively. From the boxplots we can see that the centers of the boxplots of V1, V2, V3, P2, P3 are all around -0.2, except that the center of the boxplot for P1 is around 0. In Figure 2.20 (Dash lines: high for V3, middle for V1, low for V2. Solid lines: high for P1, middle for P2, low for P3), we can also see the same pattern from the pdfs of the mean value of lower peak for V1, V2, V3, P1, P2, P3.

According to Section 2.2, given the time position vector of the maximum of higher peaks: $\mathbf{M} = (m_1, m_2, m_3, m_4, \cdots)^T$, we also consider its position variation vector:

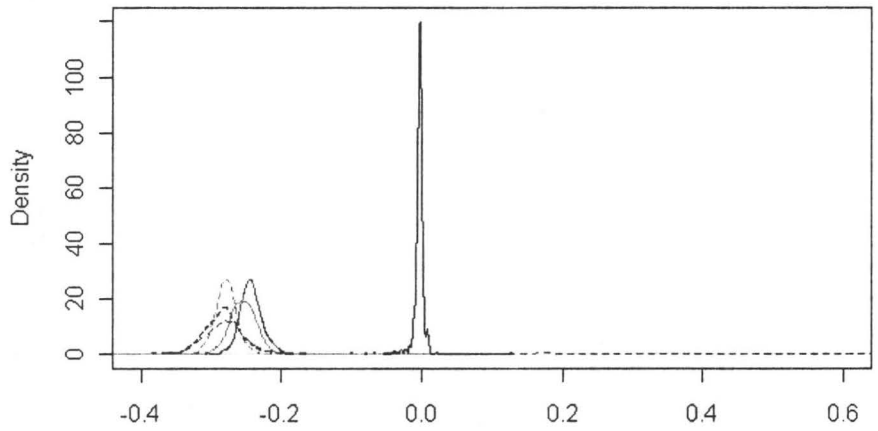


Figure 2.20: Density of the mean value of lower peak (Solid lines: Patients, Dash Lines: Volunteers).

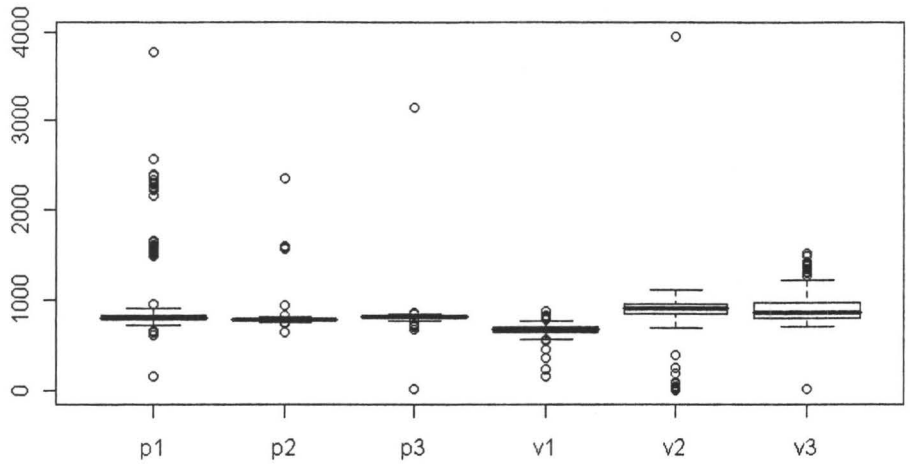


Figure 2.21: Boxplot of position variation vector of maximum.

$\mathbf{VM} = (vm_1, vm_2, vm_3, \cdots)^T$ $(vm_i = m_{i+1} - m_i,\ i = 1, 2, 3, \cdots)$ for all the electrocardiogram data of V1, V2, V3, P1, P2 and P3.

Figures 2.21-2.22 are the boxplots and pdfs of position variation vector of the electrocardiogram data: $\mathbf{VM} = (vm_1, vm_2, vm_3, \cdots)^T$ for V1, V2, V3, P1, P2, P3, respectively. In Figure 2.27 (Dash lines: high for V1, middle for V2, low for V3. Solid lines: high for P2, middle for P3, low for P1), we can see that the pdfs of position variation vector for P1, P2 and P3 are much more compact (less variance) than that of V1, V2, and V3 . In Figure 2.22, the centers of the boxplots of V1, V2, V3, P1, P2, P3 are almost the same around 800.

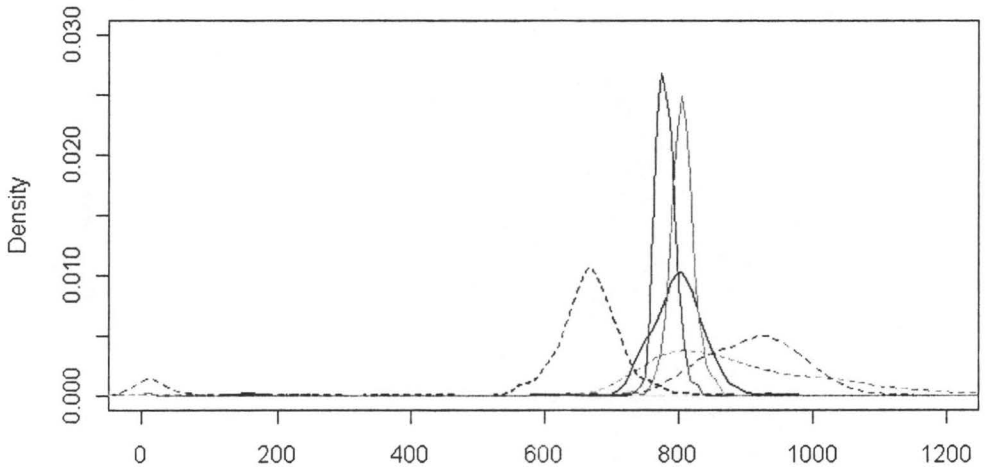


Figure 2.22: Density of position variation vector of maximum (Solid lines: Patients, Dash Lines: Volunteers).

2.3.2 Feature Vectors for Electrocardiogram Data

Consider now the means of each univariate feature discussed in Section 2.3.1. Specifically, for each subject consider the following 7 statistics,

$\overline{MH}$: mean value of maximum value of higher peak,

$\overline{RH}$: mean value of range of higher peak,

$\overline{ML}$: mean value of maximum value of lower peak,

$\overline{RL}$: mean value of range of lower peak,

$\overline{WH}$: mean value of width of higher peak,

$\overline{WL}$: mean value of width of lower peak,

$\overline{AVL}$: mean value of average value of lower peak.

Then for each subject define $\mathbf{YC}$ as the 7-component feature vector with these statistics,

$$\mathbf{YC} = (\overline{MH}, \overline{RH}, \overline{ML}, \overline{RL}, \overline{WH}, \overline{WL}, \overline{AVL})^T.$$

For the electrocardiogram data of patients and volunteers, the feature vectors will be denoted as $\mathbf{YC}_{p1}, \mathbf{YC}_{p2}, \mathbf{YC}_{p3}, \mathbf{YC}_{v1}, \mathbf{YC}_{v2}$, and $\mathbf{YC}_{v3}$.

Table 2.3 gives the value of all the seven feature vectors of the electrocardiogram data for all the patients and volunteers.

Table 2.3: The feature vectors for electrocardiogram data of the patients and volunteers.

	$\overline{MH}$	$\overline{RH}$	$\overline{ML}$	$\overline{RL}$	$\overline{WH}$	$\overline{WL}$	$\overline{AVL}$
V1	1.3213	1.9370	-0.0014	0.6143	0.05780	0.6094	-0.2848
V2	1.3145	1.8893	0.2416	0.8159	0.0857	0.7516	-0.2239
V3	0.8774	1.3201	0.0155	0.4581	0.0771	0.8111	-0.2764
P1	0.3529	0.6675	0.2167	0.5313	0.0751	0.8014	-0.0032
P2	0.4571	0.9216	0.1348	0.5993	0.0715	0.7168	-0.2434
P3	0.7563	1.2883	0.0539	0.5859	0.0651	0.7413	-0.2495

2.3.3 Comparing Univariate Shape Features of Electrocardiogram Data

In this section, We compare the electrocardiogram of patients and volunteers through each of 7 components in the feature vectors.

Figure 2.23 displays the mean of the maximum value of the higher peak and lower peak between the two group of volunteers and patients. We can see that volunteers and patients separate well in the first case. However, there is no separation in the second case. In fact, they almost overlap each other on the first point, although they separate well in the second and third point.

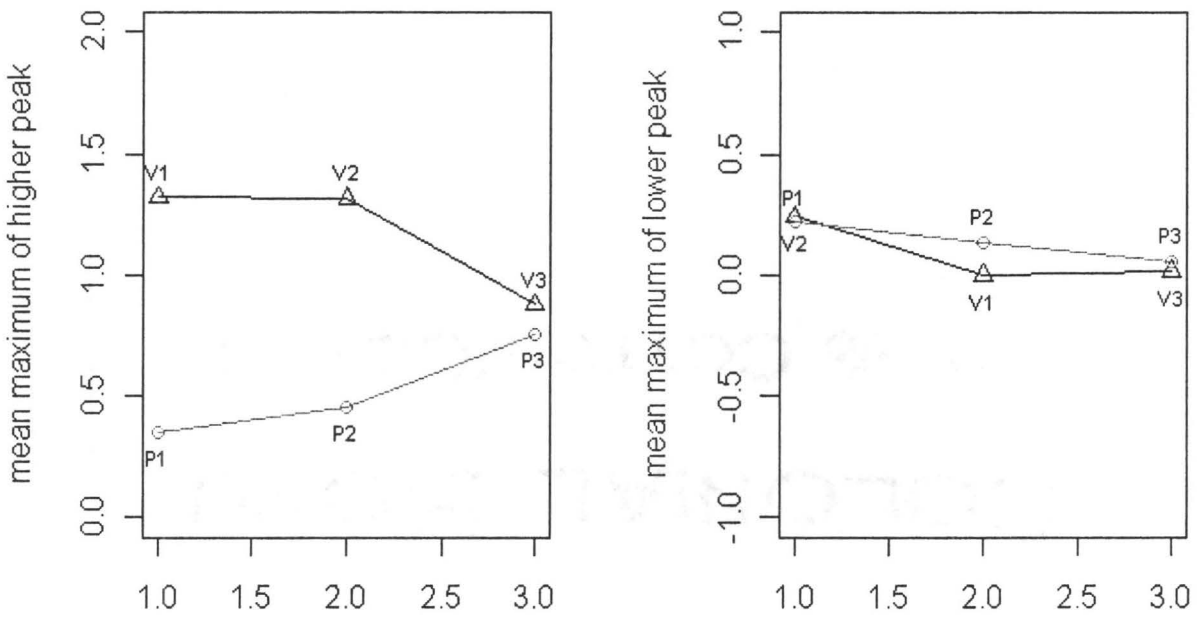


Figure 2.23: Mean of the maximum value of the higher peak (left) and lower peak (right) (Triangle: Volunteer, Circle: Patient).

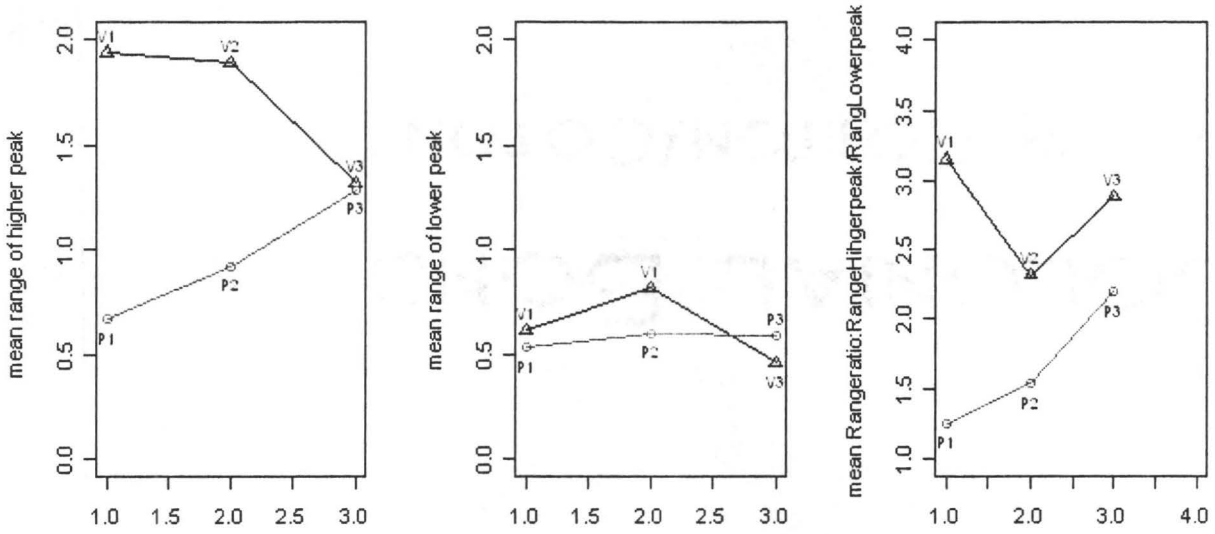


Figure 2.24: Mean of the range of the higher peak (left) and lower peak (middle), and their ratio (right) (Triangle: Volunteer, Circle: Patient).

Figure 2.24 displays the mean of the range of the higher peak and lower peak for volunteers and patients, also displays the ratio of the mean of the higher peak over the mean of the range of the lower peak for volunteers and patients. We can see that the two groups of volunteers and patients are separate well in the first case, but they are across each other in the second case. The two groups of volunteers and patients separate well in the third case.

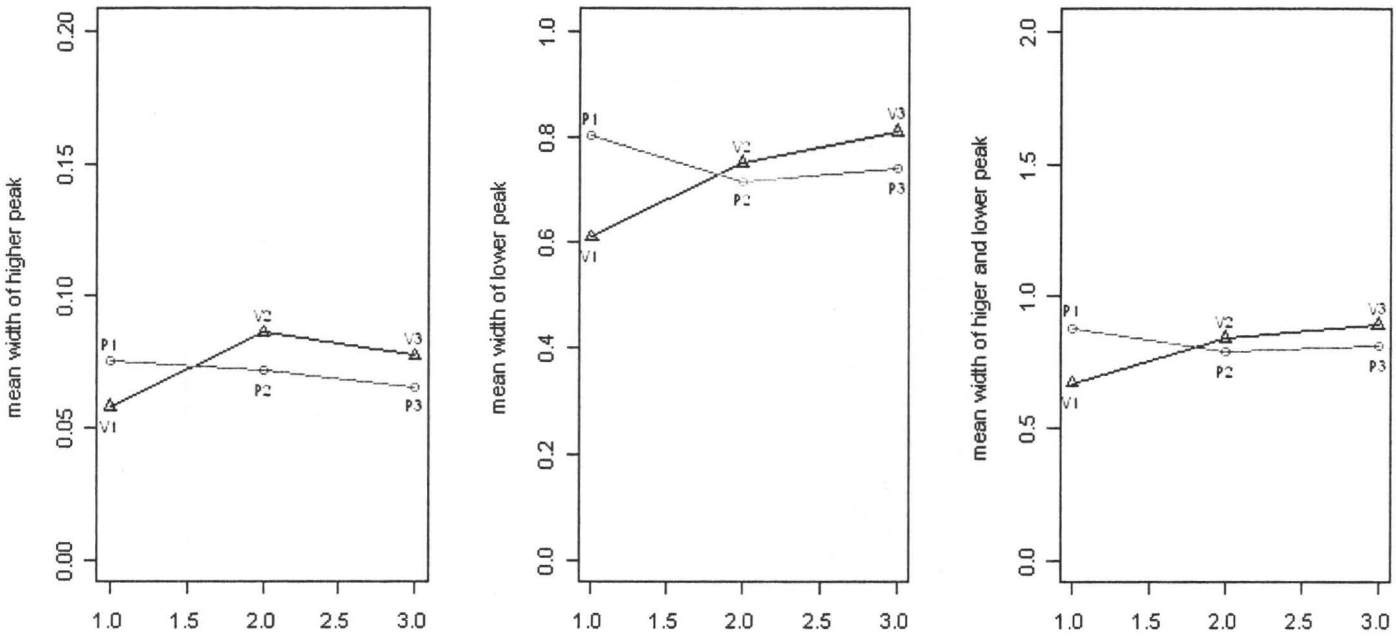


Figure 2.25: Mean of the time interval of the higher peak (left) and lower peak (middle), and their summation (right) (Triangle: Volunteer, Circle: Patient).

Figure 2.25 displays the mean of the time interval of the higher peak and lower peak for the volunteers and patients, also the sum of the mean of the higher peak and the mean of the range of the lower peak for volunteers and patients. We can see that the two groups of volunteers and patients are across each other in all the three cases.

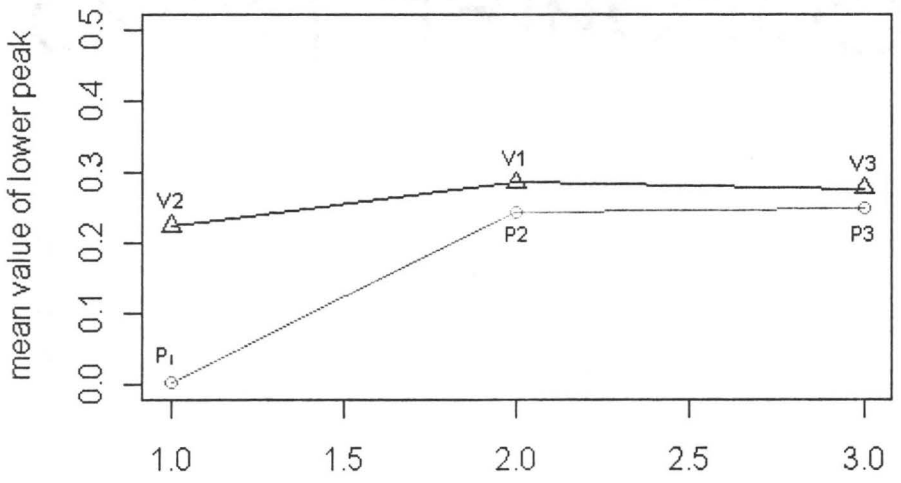


Figure 2.26: Mean of the original value of the lower peak (Triangle: Volunteer, Circle: Patient).

Figure 2.26 displays the mean of the original points value of the lower peak for volunteers and patients. We can see that the two groups of volunteers and patients separate well in this case.

Chapter 3

Clustering Analysis of Electrocardiogram Data

3.1 Hierarchical Clustering Analysis

Cluster analysis aims to group multivariate observations into subsets of similar characteristics. Traditionally, this has been accomplished through a similarity measure such as a distance to establish when two observations are close or far apart.

The hierarchical clustering (Johnson, 1967) is a natural and simple unsupervised clustering algorithm. Given a group of N data vectors $\{\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \cdots, \mathbf{X}_N\}$ to be clustered, and an $N*N$ distance matrix $D_X^{(0)}=\{\|\mathbf{X}_i - \mathbf{X}_j\|, i, j \in \{1, 2, \cdots, N\}\}$ ($\|\mathbf{X}_i - \mathbf{X}_j\|$ is the distance measure between data vector $\mathbf{X}_i$ and data vector $\mathbf{X}_j$), the basic process of this hierarchical clustering algorithm is the following:

1. *Assign each vector in* $\{\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3, \cdots, \boldsymbol{X}_N\}$ *to a cluster with each cluster contain-*

ing just one data vector. Let $\{A_1^{(0)}, A_2^{(0)}, A_3^{(0)}, \cdots, A_N^{(0)}\}$ *be the set of clusters. Calculate the* $N * N$ *distance matrix* $D_A^{(0)}=\{\left\|A_i^{(0)} - A_j^{(0)}\right\|,\ i,j \in \{1,2,\cdots,N\}\}$ ($\left\|A_i^{(0)} - A_j^{(0)}\right\|$ *is the distance measure between cluster* $A_i^{(0)}$ *and cluster* $A_j^{(0)}$). *In the case of initial step,* $D_A^{(0)} = D_X^{(0)}$.

2. Find the closest pair of clusters according to the $N*N$ *distance matrix* $D_A^{(0)}=\{\left\|A_i^{(0)} - A_j^{(0)}\right\|,\ i,j \in \{1,2,\cdots,N\}\}$ *and merge them into a single cluster, then the number of clusters in* $\{\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3, \cdots, \boldsymbol{X}_N\}$ *reduces to* $N-1$ *clusters:* $\{A_1^{(1)}, A_2^{(1)}, A_3^{(1)}, \cdots, A_{N-1}^{(1)}\}$.

3. Compute the $(N-1) * (N-1)$ *distance matrix* $D_A^{(1)}=\{\left\|A_i^{(1)} - A_j^{(1)}\right\|,\ i,j \in \{1,2,\cdots,N-1\}\}$ ($\left\|A_i^{(1)} - A_j^{(1)}\right\|$ *is the distance measure between cluster* $A_i^{(1)}$ *and cluster* $A_j^{(1)}$) *for the* $N-1$ *clusters:* $\{A_1^{(1)}, A_2^{(1)}, A_3^{(1)}, \cdots, A_{N-1}^{(1)}\}$.

4. Repeat steps 2 and 3 until all data vectors $\{\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3, \cdots, \boldsymbol{X}_N\}$ *are clustered into a single cluster of size* N.

In step 3 of the algorithm, if the distance between one cluster and another cluster $\left\|A_i^{(1)} - A_j^{(1)}\right\|$ is equal to the shortest distance from any member of cluster $A_i^{(1)}$ to any member of the other cluster $A_j^{(1)}$, we say that it is *single-linkage hierarchical clustering.*

Alternatively, in step 3 of the algorithm, if the distance $\left\|A_i^{(1)} - A_j^{(1)}\right\|$ between one cluster and another cluster is equal to the greatest distance from any member of one cluster $A_i^{(1)}$ to any member of the other cluster $A_j^{(1)}$, then it is called *complete-linkage hierarchical clustering.*

Also, in step 3 of the algorithm, if the distance $\left\|A_i^{(1)} - A_j^{(1)}\right\|$ between one cluster and another cluster is equal to the distance from the centroid $\mathbf{C}_i^{(1)}$of one cluster $A_i^{(1)}$ to the centroid $\mathbf{C}_j^{(1)}$ of the other cluster $A_j^{(1)}$, then it is called *centroid-linkage hierarchical*

clustering.

In clustering the six feature vectors: $\{\mathbf{YC}_{p1}, \mathbf{YC}_{p2}, \mathbf{YC}_{p3}, \mathbf{YC}_{v1}, \mathbf{YC}_{v2}, \mathbf{YC}_{v3}\}$ for the electrocardiogram data of patients and volunteers, we use centroid-linkage hierarchical clustering in step 3. The distance measure $\left\|\mathbf{C}_j^{(1)} - \mathbf{C}_i^{(1)}\right\|$ in the centroid-linkage hierarchical clustering method is Euclid distance measure, and the centroid $\mathbf{C}_i^{(1)}$ of cluster $A_i^{(1)}$ is calculated as the average of the data vectors in the corresponding ith cluster partition $A_i^{(1)}$:

$$\mathbf{C}_i^{(1)} = \frac{1}{N_i} \sum_{j=1}^{N_i} \mathbf{X}_j^{(1)},$$

where N_i denotes the number of data vectors in $A_i^{(1)}$ and $X_j^{(1)} \in A_i^{(1)}$, $j = 1, \cdots, N_i$.

Table 3.1: Hierarchical clustering process for electrocardiogram data.

	Cluster Partitions
Initial	$\{\mathbf{YC}_{p1}\}$, $\{\mathbf{YC}_{p2}\}$, $\{\mathbf{YC}_{p3}\}$, $\{\mathbf{YC}_{v1}\}$, $\{\mathbf{YC}_{v2}\}$, $\{\mathbf{YC}_{v3}\}$
1^{st} step	$\{\mathbf{YC}_{p1}\}$, $\{\mathbf{YC}_{p2}\}$, $\{\mathbf{YC}_{v1}\}$, $\{\mathbf{YC}_{v2}\}$, $\{\mathbf{YC}_{p3}, \mathbf{YC}_{v3}\}$
2^{nd} step	$\{\mathbf{YC}_{v1}\}$, $\{\mathbf{YC}_{v2}\}$, $\{\mathbf{YC}_{p3}, \mathbf{YC}_{v3}\}$, $\{\mathbf{YC}_{p1}, \mathbf{YC}_{p2}\}$
3^{rd} step	$\{\mathbf{YC}_{p3}, \mathbf{YC}_{v3}\}$, $\{\mathbf{YC}_{p1}, \mathbf{YC}_{p2}\}$, $\{\mathbf{YC}_{v1}, \mathbf{YC}_{v2}\}$
4^{th} step	$\{\mathbf{YC}_{p3}, \mathbf{YC}_{v3}, \mathbf{YC}_{p1}, \mathbf{YC}_{p2}\}$, $\{\mathbf{YC}_{v1}, \mathbf{YC}_{v2}\}$
5^{th} step	$\{\mathbf{YC}_{p3}, \mathbf{YC}_{v3}, \mathbf{YC}_{p1}, \mathbf{YC}_{p2}, \mathbf{YC}_{v1}, \mathbf{YC}_{v2}\}$

Table 3.1 displays the process of hierarchical clustering. Figure 3.1 gives more details in the visualized sense for centroid-linkage hierarchical clustering for the six feature

vectors: $\{\mathbf{YC}_{p1}, \mathbf{YC}_{p2}, \mathbf{YC}_{p3}, \mathbf{YC}_{v1}, \mathbf{YC}_{v2}, \mathbf{YC}_{v3}\}$ for the electrocardiogram data of patients and volunteers.

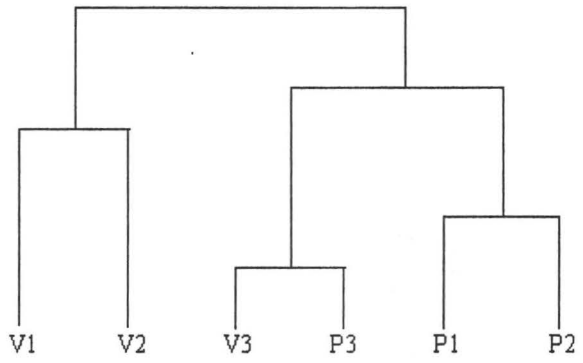


Figure 3.1: Hierarchical clustering tree for electrocardiogram data.

From Table 3.1 and Figure 3.1 we can see that the feature vectors for V1 and V2 are close to each other, the feature vectors for V3 and P3 are close to each other, and the feature vectors for P1 and P2 are close to each other. We know that all the six electrocardiogram data come from two clusters: patients and volunteers. From Figure 3.1 of centroid-linkage hierarchical clustering, we can see that P1, P2, P3, V3 are in one cluser, and V1, V2 are in the other cluster.

3.2 K-means Clustering Analysis

Another solution to the well-known clustering problem is provided by the K-means clustering method (MacQueen, 1967), which has been established as a good and simple unsupervised learning algorithm. We wish to classify a group of N data vectors

$\{\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \cdots, \mathbf{X}_N\}$ into a certain number of clusters (assume K clusters, which is known a priori) with cluster centroids $\{\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3, \cdots, \mathbf{C}_K\}$. Initially, we should give K centroids: $\{\mathbf{C}_1^{init}, \mathbf{C}_2^{init}, \mathbf{C}_3^{init}, \cdots, \mathbf{C}_K^{init}\}$, one for each cluster. The next step is to take each data vector in $\{\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \cdots, \mathbf{X}_N\}$ and associate it to the nearest centroid in the centroid sets: $\{\mathbf{C}_1^{init}, \mathbf{C}_2^{init}, \mathbf{C}_3^{init}, \cdots, \mathbf{C}_K^{init}\}$. The ith ($i = 1, \cdots, K$) cluster partition is $\{\mathbf{X}_1^{(i)}, \cdots, \mathbf{X}_{N_i}^{(i)}\}$, where N_i is the number of data vectors in ith cluster partition. We also have the simple relationship: $N_1 + \cdots + N_K = N$. Then according to the new K clusters partition, we recalculate the new K centroids: $\{\mathbf{C}_1^{new}, \mathbf{C}_2^{new}, \mathbf{C}_3^{new}, \cdots, \mathbf{C}_K^{new}\}$. After we get the new K centroids: $\{\mathbf{C}_1^{new}, \mathbf{C}_2^{new}, \mathbf{C}_3^{new}, \cdots, \mathbf{C}_K^{new}\}$, we repeat the process of taking each data vector in $\{\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \cdots, \mathbf{X}_N\}$ and associate it to the nearest centroid in $\{\mathbf{C}_1^{new}, \mathbf{C}_2^{new}, \mathbf{C}_3^{new}, \cdots, \mathbf{C}_K^{new}\}$ and recalculate the new K centroids again. As a result of this loop we will notice that the K centroids change their location step by step until no more changes are done, then we say that data vectors $\{\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \cdots, \mathbf{X}_N\}$ are classified into K clusters.

In general, the K-means clustering method is actually used to minimize the square error function of the classification, which is called the objective function:

$$J = \sum_{i=1}^{K} \sum_{j=1}^{N_i} \left\| \mathbf{X}_j^{(i)} - C_i \right\|^2,$$

where $\left\| \mathbf{X}_j^{(i)} - \mathbf{C}_i \right\|$ is a chosen distance measure between the data vector $\mathbf{X}_j^{(i)}$ and its cluster center vector $\mathbf{C}_i$. We can see that the objective function is a measure of the distance of all the N data vectors $\{\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \cdots, \mathbf{X}_N\}$ from their respective cluster

centroids $\{\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3, \cdots, \mathbf{C}_K\}$. From the recursive procedure of K-means clustering method, we know that we will reduce the value of the objective function J when we get the new K centroids: $\{\mathbf{C}_1^{new}, \mathbf{C}_2^{new}, \mathbf{C}_3^{new}, \cdots, \mathbf{C}_K^{new}\}$, and we also know that $J \geq 0$. So if good initial K centroids $\{\mathbf{C}_1^{init}, \mathbf{C}_2^{init}, \mathbf{C}_3^{init}, \cdots, \mathbf{C}_K^{init}\}$ are given (Bradley and Fayyad, 1998), then the K-means clustering algorithm will converge in finite steps. A safe choice is to place the initial K centroids $\{\mathbf{C}_1^{init}, \mathbf{C}_2^{init}, \mathbf{C}_3^{init}, \cdots, \mathbf{C}_K^{init}\}$ as far away as possible from each other.

The algorithm is composed of the following steps:

1. Place K initial centroids $\{\mathbf{C}_1^{init}, \mathbf{C}_2^{init}, \mathbf{C}_3^{init}, \cdots, \mathbf{C}_K^{init}\}$ *into the space represented by the data vectors that are being clustered* $\{\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \cdots, \mathbf{X}_N\}$.

2. Assign each data vector in $\{\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \cdots, \mathbf{X}_N\}$ *to the cluster partition that has the closest centroid in* $\{\mathbf{C}_1^{init}, \mathbf{C}_2^{init}, \mathbf{C}_3^{init}, \cdots, \mathbf{C}_K^{init}\}$.

3. When all data vectors in $\{\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \cdots, \mathbf{X}_N\}$ *have been assigned, recalculate the new K centroids* $\{\mathbf{C}_1^{new}, \mathbf{C}_2^{new}, \mathbf{C}_3^{new}, \cdots, \mathbf{C}_K^{new}\}$.

4. Repeat Steps 2 and 3 until the centroids no longer move or the value of the objective function J is less than some preset valve ε.

The distance measure $\left\|\mathbf{X}_j^{(i)} - \mathbf{C}_i\right\|$ for the K-means clustering method is Euclid distance measure, and the new K centroids $\{\mathbf{C}_1^{new}, \mathbf{C}_2^{new}, \mathbf{C}_3^{new}, \cdots, \mathbf{C}_K^{new}\}$ are calculated as the average center of the corresponding ith $(i = 1, \cdots, K)$ cluster partition $\{\mathbf{X}_1^{(i)}, \cdots, \mathbf{X}_{N_i}^{(i)}\}$:

$$\mathbf{C}_i^{new} = \frac{1}{N_i} \sum_{j=1}^{N_i} \mathbf{X}_j^{(i)}, \quad i = 1, \cdots, K.$$

Table 3.2: Results for the process of K-means clustering for the electrocardiogram data.

Initial step	$\mathbf{C}_{10}$ =(0.5, 0.9, 0.1, 0.6, 0.03, 0.7, −0.1) $\mathbf{C}_{20}$ =(0.2, 0.6, 0.2, 0.5, 0.01, 0.90)
1^{st} step	Cluster 1={V1, V2, V3, P1, P3} $\mathbf{C}_{11}$=(0.945, 1.47, 0.09, 0.615, 0.07, 0.726, −0.256) Cluster 2={P2} $\mathbf{C}_{21}$=(0.353, 0.667, 0.217, 0.531, 0.0751, 0.801, −0.003)
2^{nd} step	Cluster 1={V1, V2, V3, P3} $\mathbf{C}_{12}$= (1.068, 1.609, 0.077, 0.619, 0.0714, 0.728, −0.259) Cluster 2={P1, P2} $\mathbf{C}_{22}$=(0.405, 0.795, 0.176, 0.565, 0.073, 0.759, −0.123)
3^{rd} step	Cluster 1={V1, V2, V3, P3} $\mathbf{C}_{13}$= (1.068, 1.609, 0.077, 0.619, 0.0714, 0.728, −0.259) Cluster 2={P1, P2} $\mathbf{C}_{23}$=(0.405, 0.795, 0.176, 0.565, 0.073, 0.759, −0.123)

According to Section 2.3, for the electrocardiogram data, our objective is equivalent to classify a group of 6 feature data vectors $\{\mathbf{YC}_{p1}, \mathbf{YC}_{p2}, \mathbf{YC}_{p3}, \mathbf{YC}_{v1}, \mathbf{YC}_{v2}, \mathbf{YC}_{v3}\}$ into two clusters with cluster centroids $\{\mathbf{C}_1, \mathbf{C}_2\}$. Table 3.2 shows the process of the K-

means clustering algorithm for the electrocardiogram data. The clusters and centroids are given in every recursive step in the K-means clustering algorithm. The K-means algorithm stops at the 3^{rd} step, as the centroids do not move anymore.

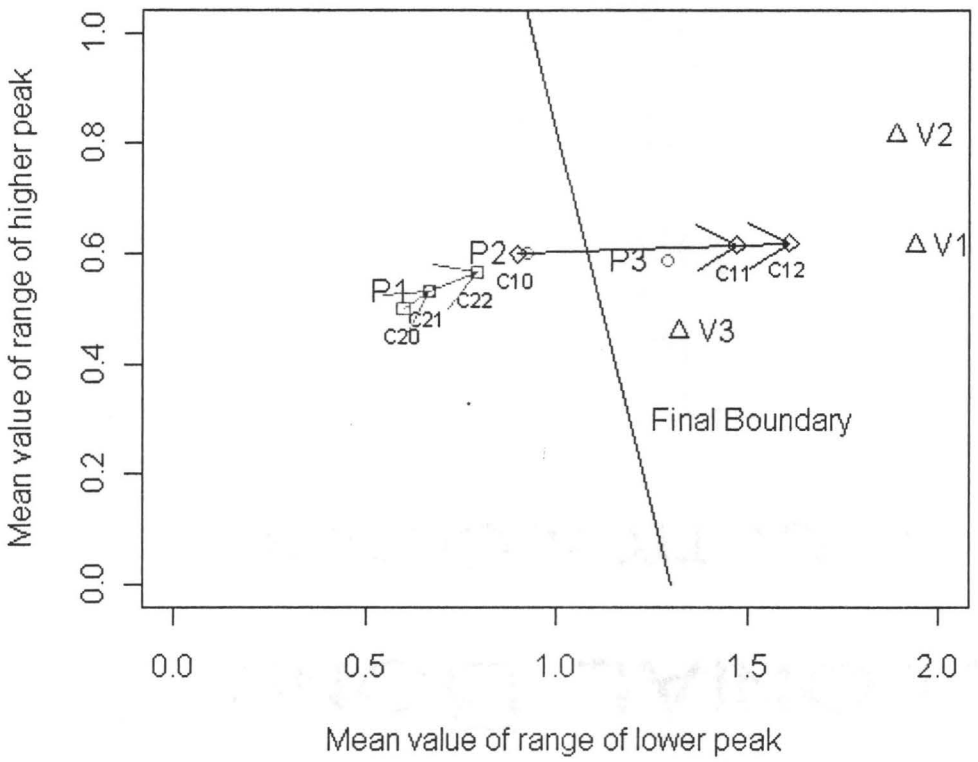


Figure 3.2: K-means clustering for electrocardiogram data.

Figure 3.2 gives more details in a visualized manner. The mean value of range of lower peak is used as the x-coordinate, and the mean value of range of higher peak is used as the y-coordinate for the figure. The triangle points represent the feature vectors of the electrocardiogram data for healthy volunteers: V1, V2, V3. The circle points

represent the feature vectors for patients: P1, P2, P3. The square points represent the centroids in every recursive step of the K-means clustering algorithm. And the arrows show the direction from one step to the next step in the process of K-means clustering. The straight line in the middle of the figure is the final separation of the 6 feature vectors into two clusters.

From Figure 3.2 of K-means clustering method for the case of electrocardiogram data of patients and volunteers, we can see that V1, V2, V3 and P3 are in one cluster, and P1, P2 are in the other cluster.

3.3 Andrews-Plot Clustering Analysis

Plotting has been one of the most useful statistical tools in data analysis, especially in exploratory data analysis of high-dimensional data. It is well-known that the plotting of residuals is a reliable way to test the adequacy of a model fitting, and distributional assumptions are frequently based on probability plots. Some plotting techniques such as Trellis plots and parallel coordinate plots were found in Wegman and Carr (1993) and Wegman et al. (1993). But Andrews plot introduce by Andrewss (1972) stands out, as it is supported by solid mathematical justification and has the desirable property of preserving means, distances and variances of the original vector data. Embrechts and Herzberg (1991) introduced other orthogonal functions such as Chebychev polynomials and Legendre polynomials into Andrews plot and they show their good performance in clustering the Iris Data (Fisher, 1936). Wavelets were also introduced into Andrews plot

by Embrechts et al. (1995). Khattree and Naik (2002) provided other trigonometric functions for Andrews plot.

3.3.1 Clustering Analysis Using the First Andrews Plot Function

Given a data vector $\mathbf{Y} = (y_1, y_2, y_3, \cdots)^T$, the Andrews-Plot function for $\mathbf{Y}$ is:

$$A_{\mathbf{Y}}(t) = y_1/\sqrt{2} + y_2 \sin t + y_3 \cos t + y_4 \sin 2t + y_5 \cos 2t + \cdots, -\pi \leq t \leq \pi. \quad (3.1)$$

The Andrews-Plot function has many desirable properties relevant to the clustering analysis for multivariate data, especially for high-dimensional data.

1. Andrews plot function preserves means of the data

Let $\overline{\mathbf{Y}}$ be the mean of a group data vectors $\{\mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3, \cdots, \mathbf{Y}_N\}$, then it is obvious that:

$$A_{\overline{\mathbf{Y}}}(t) = \frac{1}{N} \sum_{i=1}^{N} A_{\mathbf{Y}_i}(t),$$

which means that the Andrews plot of the average vector $\overline{\mathbf{Y}}$ is the same as the average of the Andrews plots for all the original data vectors: $\{\mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3, \cdots \mathbf{Y}_N\}$.

2. Andrews plot function preserves distances of the data

Assume we have two N-dimensional data vectors: $\mathbf{Y} = (y_1, y_2, y_3, \cdots, y_N)^T$ and $\mathbf{X} = (x_1, x_2, x_3, \cdots, x_N)^T$. $\|\mathbf{X} - \mathbf{Y}\|$ is the Euclidean distance between them. We define the distance between the Andrews plot functions of $\mathbf{X}$ and $\mathbf{Y}$ as the following:

$$\|A_{\mathbf{X}}(t) - A_{\mathbf{Y}}(t)\|_{L_2} = \int_{-\pi}^{\pi} (A_{\mathbf{X}}(t) - A_{\mathbf{Y}}(t))^2 dt.$$

Moreover, we can see from the following that this distance is proportional to the Euclidean distance $\|\mathbf{X} - \mathbf{Y}\|$.

$$\begin{aligned}
&\|A_{\mathbf{X}}(t) - A_{\mathbf{Y}}(t)\|_{L_2} \\
&= \textstyle\int_{-\pi}^{\pi} (A_{\mathbf{X}}(t) - A_{\mathbf{Y}}(t))^2 dt \\
&= \tfrac{1}{2}(x_1 - y_1)^2 \textstyle\int_{-\pi}^{\pi} 1 dt + (x_2 - y_2)^2 \int_{-\pi}^{\pi} (\sin t)^2 dt + (x_3 - y_3)^2 \int_{-\pi}^{\pi} (\cos t)^2 dt + \cdots \\
&= \pi \sum_{i=1}^{N} (x_i - y_i)^2
\end{aligned}$$

The last equality comes from $\int_{-\pi}^{\pi} 1 dt = 2\pi$, and $\int_{-\pi}^{\pi} (\sin t)^2 dt = \int_{-\pi}^{\pi} (\cos t)^2 dt = \pi$.

So we can say that if the original data vectors $\mathbf{X}$ and $\mathbf{Y}$ are close in the vector space, then the Andrews plot functions $A_{\mathbf{X}}(t)$ and $A_{\mathbf{Y}}(t)$ will still stay close. Then clusters and outliers in the original data vectors can be identified visually from the respective Andrews plot functions.

3. Andrews plot function preserves variance of the data

Assume we have a data vector: $\mathbf{Y} = (y_1, y_2, y_3, \cdots, y_N)^T$, and $\{y_1, y_2, y_3, \cdots, y_N\}$ are uncorrelated random variables with common variance σ^2.

We have $var(A_{\mathbf{Y}}(t)) = \sigma^2(1/2 + (\sin t)^2 + (\cos t)^2 + (\sin 2t)^2 + (\cos 2t)^2 + \cdots)$. Thus when N is odd, $var(A_{\mathbf{Y}}(t)) = \frac{1}{2}N\sigma^2$, and when N is even, $var(A_{\mathbf{Y}}(t)) = \frac{1}{2}(N - 1 + 2\sin^2(Nt/2))\sigma^2$. And we have the following relationship,

$$\frac{1}{2}(N-1)\sigma^2 \leq var(A_{\mathbf{Y}}(t)) \leq \frac{1}{2}(N+1)\sigma^2.$$

In the first case (when N is odd) the variance does not depend on t and in the second case (when N is even) the relative dependence on t is small and it decreases when N increases. Thus, the variability of Andrews plot function is almost constant across the graph.

4. The Andrews plot function produces one-dimensional projections

Given a data vector: $\mathbf{Y} = (y_1, y_2, y_3, \cdots)^T$ and its Andrews plot function $A_{\mathbf{Y}}(t) = y_1/\sqrt{2} + y_2 \sin t + y_3 \cos t + y_4 \sin 2t + y_5 \cos 2t + \cdots$, and a particular value of $t = t_0$, we have the following,

$$A_{\mathbf{Y}}(t_0) = \frac{\mathbf{Y}^T \mathbf{A}(t_0)}{\mathbf{A}^T(t_0)\mathbf{A}(t_0)} \mathbf{A}^T(t_0)\mathbf{A}(t_0),$$

where $\mathbf{A}(t_0) = (1/\sqrt{2}, \sin t_0, \cos t_0, \sin 2t_0, \cos 2t_0, \cdots)^T$, $\mathbf{Y}^T$ and $\mathbf{A}^T(t_0)$ are the transpose of $\mathbf{Y}$ and $\mathbf{A}(t_0)$, respectively. As we know that $\frac{\mathbf{Y}^T \mathbf{A}(t_0)}{\mathbf{A}^T(t_0)\mathbf{A}(t_0)}$ is the length of the projection of $\mathbf{Y}$ on the vector $\mathbf{A}(t_0)$, and $\mathbf{A}^T(t_0)\mathbf{A}(t_0)$ is a constant, so $\mathbf{A}_{\mathbf{Y}}(t_0)$ is proportional to the length of the projection of $\mathbf{Y}$ on the vector $\mathbf{A}(t_0)$. Hence, clusterings, outlier pattern, or other peculiarities may be revealed in the projection on this one-dimensional space more clearly. The advantage of Andrews plot is that we plot a continuum of many such nice one-dimensional projections on the graph.

Figure 3.3 displays Andrews plots using formula (3.1) for the six feature data vectors $\{\mathbf{YC}_{p1}, \mathbf{YC}_{p2}, \mathbf{YC}_{p3}, \mathbf{YC}_{v1}, \mathbf{YC}_{v2}, \mathbf{YC}_{v3}\}$ of the electrocardiogram data.

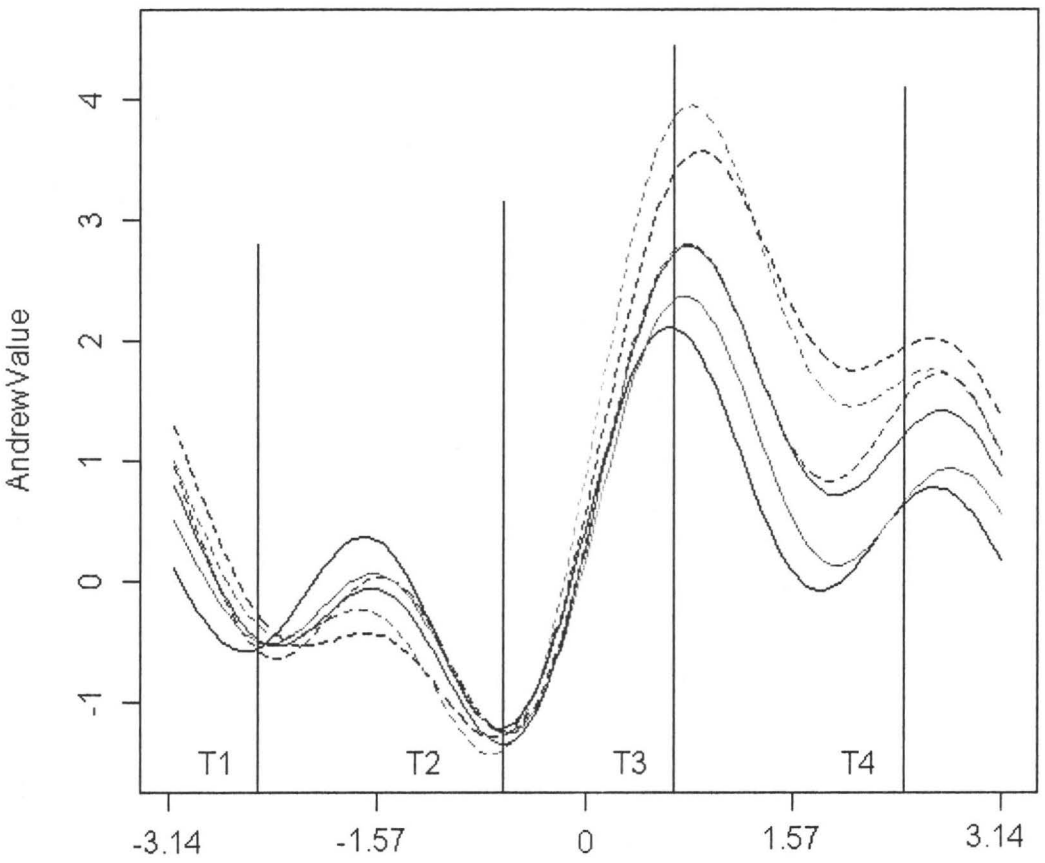


Figure 3.3: Andrews plot using formula (3.1) for electrocardiogram data for all volunteers and patients (Dashed Lines: Volunteer, Solid Lines: Patient).

It can be seen from Figure 3.3 that at time point T_3, the most top dashed line represents the Andrews plot for V2, and the next top dashed line represents the Andrews plot for V1, and the lowest dashed line represents the Andrews plot for V3, and the most top solid line represents the Andrews plot for P3, and the next solid line represents the Andrews plot for P2, and the lowest solid line represents the Andrews plot for P1. Figure 3.3 reveals that the Andrews plots of V1 and V2 are close to each other and show a similar pattern, the Andrews plots of P1 and P2 are close to each other and show also a similar pattern. The Andrews plots of V1 and V2 are bellow the Andrews plots of P1 and P2 between T_1 and T_2, but the Andrews plots of V1 and V2 are top on the Andrews plots of P1 and P2 on all other slots. So it is easy to see that V1 and V2 are in one cluster, and P1 and P2 are in another cluster. As for the Andrews plots of V3 and P3, they are very close to each other, and they even coincide between T_2 and T_4. We can also see that the Andrews plots of V3 and P3 always stay in the middle of the plots of V1 and V2 and the plots of P1 and P2. But at time T_1, we can see that all the three solid lines of P1, P2, P3 pass through the same point, and at time T_4, we can see that all the three dashed lines of V1, V2, and V3 are very close to each other, separated from the solid lines of P1, P2, and P3. So we conclude from the above Andrews plots using formula (3.1) to classifying the 6 data vectors $\{\mathbf{YC}_{p1}, \mathbf{YC}_{p2}, \mathbf{YC}_{p3}, \mathbf{YC}_{v1}, \mathbf{YC}_{v2}, \mathbf{YC}_{v3}\}$ that V1, V2, V3 are in the same cluster, P1, P2 and P3 are in the same cluster.

3.3.2 Clustering Analysis Using the Second Andrews Plot Function

Unfortunately, all even numbered terms in the Andrews plot function (3.1) will simultaneously vanish when $t = 0$, thereby only the features of the odd number terms are present on the graph. It is also similar for all odd numbered terms when t is a multiple of $\pi/2$. Khattree and Naik (2002) provided another trigonometric functions so that many of these terms do not simultaneously vanish at any given t, as they do for the Fourier series in the original Andrews plot function.

Given a data vector $\mathbf{Y} = (y_1, y_2, y_3, \cdots)^T$, then a second Andrews plot function for the data vector $\mathbf{Y}$ has been proposed and is given by:

$$B_{\mathbf{Y}}(t) = \frac{1}{\sqrt{2}}(y_1 + y_2(\sin t + \cos t) + y_3(\sin t - \cos t) + y_4(\sin 2t + \cos 2t) + \cdots), -\pi \leq t \leq \pi. \tag{3.2}$$

Note that the addition and substraction of terms of functions $\sin(jt)$ and $\cos(jt)$ result in every element in the high-dimensional data vector $\mathbf{Y}$ being exposed to a sine function as well as a cosine function. Then the new Andrews plot function will be more informative from a statistical point of view. Also, unlike (3.1) the trigonometric terms in (3.2) do not simultaneously vanish at any given t. Furthermore, Andrews plot function (3.2) preserves all the desirable properties of Andrews plot function (3.1), which were shown in Section 3.3.1.

Figure 3.4 displays the Andrews plots using formula (3.2) for the six feature data

vectors $\{\mathbf{YC}_{p1}, \mathbf{YC}_{p2}, \mathbf{YC}_{p3}, \mathbf{YC}_{v1}, \mathbf{YC}_{v2}, \mathbf{YC}_{v3}\}$ of the electrocardiogram data.

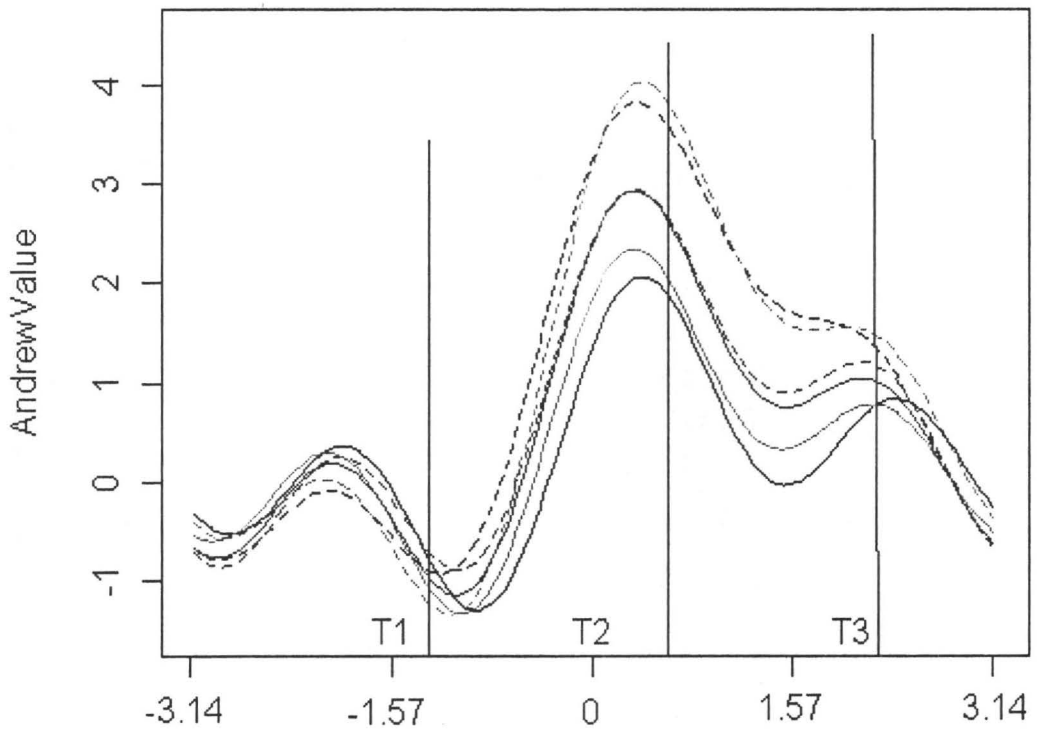


Figure 3.4: Andrews plot using formula (3.2) for electrocardiogram data for all volunteers and patients (Dashed Lines: Volunteer, Solid Lines: Patient).

It can be seen from Figure 3.4 that at time point T_2, the most top dashed line represents the Andrews plot for V2, and the next top dashed line represents the plot for V1, and the lowest dashed line represents the plot for V3. The most top solid line

represents the Andrews plot for P3, and the next top solid line represents the plot for P2, and the lowest solid line represents the plot for P1. From Figure 3.4, we can see that the Andrews plots of V1 and V2 are much closer to each other and show similar pattern, the plots of P1 and P2 are much closer to each other and show a similar pattern, compared to the figure of Andrews plot using formula (3.1). The Andrews plots of V1 and V2 are below the plots of P1 and P2 before T_1, but the Andrews plots of V1 and V2 are always top on the plots of P1 and P2 on all other slots. So it is clear that V1 and V2 are in one cluster, and P1 and P2 are in another cluster. As for the Andrews plots of V3 and P3, they are very close to each other, and even coincide between T_1 and T_2. We can also see that the Andrews plots of V3 and P3 always stay in the middle of the plots of V1 and V2 and the plots of P1 and P2. But at points around T_3, we can see that all the three dashed lines of V1, V2, and V3 are very close to each other, all the three solid lines of P1, P2, and P3 are very close to each other, and the two groups of solid lines and dashed lines are well separated from each other. So we can conclude from the above Andrews plots using formula (3.2) for 6 feature data vectors $\{\mathbf{YC}_{p1}, \mathbf{YC}_{p2}, \mathbf{YC}_{p3}, \mathbf{YC}_{v1}, \mathbf{YC}_{v2}, \mathbf{YC}_{v3}\}$ that V1, V2, V3 are in one cluster, and P1, P2 and P3 are in another cluster.

Chapter 4

Conclusions and Future Study

4.1 Extracting Statistical Features for Electrocardiogram Data

This thesis focuses on the statistical analysis of electrocardiogram data, a type of data that tend to be of very high dimension. In Section 2.3 we defined seven statistics that capture interesting features of the electrocardiogram data: maximum value of higher peak, range of higher peak, maximum value of lower peak, range of lower peak, width of higher peak, width of lower peak, average value of lower peak, separately. The aim was to study these features in their own right and to use them to classify the subjects. The methods discussed were illustrated on a small but revealing set of 6 electrocardiograms: 3 of them come from cancer patients while the other 3 come from healthy volunteers. From the discussion in Section 2.3, we can see that some statistical features, when used

individually, such as maximum value of higher peak, range of higher peak, ratio of range of higher peak over range of lower peak, average value of lower peak can separate the electrocardiogram data well into two groups as patients and volunteers. But unfortunately, other statistical features such as maximum value of lower peak, range of lower peak, width of higher peak, width of lower peak and variation of position of maximum are much less effective in clustering the electrocardiogram data correctly into the two groups as patients and volunteers.

4.2 Multivariate Classification Using Feature Vectors

We then used the feature vectors in a multivariate way to perform the classification. We put all the seven import statistical features into a feature vector to represent the electrocardiogram data of all patients and volunteers: $\{\mathbf{YC}_{p1}, \mathbf{YC}_{p2}, \mathbf{YC}_{p3}, \mathbf{YC}_{v1}, \mathbf{YC}_{v2}, \mathbf{YC}_{v3}\}$. Then several clustering analysis methods were applied on these six feature vectors. In Section 3.1 on hierarchical clustering analysis, $\mathbf{YC}_{v1}$ for V1 and $\mathbf{YC}_{v2}$ for V2 are close to each other, $\mathbf{YC}_{v3}$ for V3 and $\mathbf{YC}_{p3}$ for P3 are close to each other, and $\mathbf{YC}_{p1}$ for P1 and $\mathbf{YC}_{p2}$ for P2 are close to each other. And according to the results from centroid-linkage hierarchical clustering, $\mathbf{YC}_{p1}$, $\mathbf{YC}_{p2}$, $\mathbf{YC}_{p3}$, $\mathbf{YC}_{v3}$ are in one group, and $\mathbf{YC}_{v1}$, $\mathbf{YC}_{v2}$ are in the other group. In Section 3.2 on K-means clustering analysis for the six feature vectors: $\mathbf{YC}_{v1}$ for V1, $\mathbf{YC}_{v2}$ for V2, $\mathbf{YC}_{v3}$ for V3 and $\mathbf{YC}_{p3}$ for P3 are in one cluster,

and $\mathbf{YC}_{p1}$ for P1, $\mathbf{YC}_{p2}$ for P2 are in another cluster. But we can also see that V3 and P3 are very close to each other either in Euclidean distance of feature vectors: $\mathbf{YC}_{v3}$ and $\mathbf{YC}_{p3}$ in the hierarchical clustering analysis, or in the pattern and magnitude of the original data plot for P3 and V3 in Section 2.1. The possible reason may be that cancer patient P3 is in his/her early stage of cancer and the strength of his/her heart is a little bit weaker than the normal healthy person, or volunteer V3 has no cancer, but he/she is only a little bit weak in the strength of the heart. So in K-means clustering analysis, V3 and P3 are partitioned into V1 and V2's group, but in hierarchical clustering, V3 and P3 are partitioned into P1 and P2's group. Also the clustering methods of hierarchical clustering analysis and K-means clustering analysis do not find substantive differences to separate V3 and P3.

As discussed in Section 3.3, Andrews plot has solid mathematical justification and has desirable properties such as preserving means, distances and variance of the original vector data. Clusterings, outlier pattern, or other peculiarities in the original space may be revealed in the projection to one-dimensional space more clearly, which may be otherwise obscured in higher dimensions. In Section 3.3 on the Andrews plot clustering analysis, the method resolves the difficulty found by the hierarchical clustering analysis and K-means clustering analysis in separating V3 and P3. In fact, the Andrews plots using formula (3.1) and formula (3.2) for the six feature vectors clusters $\mathbf{YC}_{v1}$ for V1, $\mathbf{YC}_{v2}$ for V2, $\mathbf{YC}_{v3}$ for V3 into one group, $\mathbf{YC}_{p1}$ for P1, $\mathbf{YC}_{p2}$ for P2 and $\mathbf{YC}_{p3}$ for P3 into another group, which is the correct classification. Note, however, that the use of Andrews plots is more empirical and requires a visual analysis of the plot. Also, it

should be pointed out that the data set is small and thus no broad conclusions can be drawn at this point.

4.3 Future Study

We look forward to the following extensions of the work done in this thesis.

- **More Data**. As noted in the thesis, a limitation of the study is the small number of electrocardiograms available at this time. As a result, no broad conclusions could be drawn. We plan to try the methods on a large number of subjects. Discussion with Dr. Raimond Wong from the Hamilton Regional Cancer Center suggests that at least 30 more electrocardiogram data will be available in the future.

- **Feature Extraction**. As discussed in Chapters 2-3 of the thesis, there are two stages in the classification methods presented: feature extraction and clustering analysis of the ensuing feature vectors. Regarding feature extraction, two approaches are used in the literature: pick appealing features driven by the data or use standard features. Our approach was to use appealing features which in part were suggested by the researchers from the cancer clinic. However standard methods include sample moments, Fourier coefficients and wavelet representations. The recent work by Epifanio (2008) gives some guidance on the use of standard methods and illustrates them with speech recognition data and biomedical data. Give the shape of electrocardiogram data, Fourier and wavelet representations may have

some potential. We plan to extend this on the existing and new electrocardiogram data.

- **Other Clustering Methods**. The literature on classification is rich. Two methods that have been shown to perform quite well for some types of data are artificial neural networks based on nonlinear models (Lippmann, et al., 1991) or fuzzy C-Means clustering (Dunn, 1973, Bezdek, 1981). These methods are more sophisticated and require more work in their implementation. It would be interesting to explore their use for our electrocardiogram data.

- **Formal Grouping**. It would be useful to have more quantitative guidance on the application of the methods. For instance, how many elements should are included in feature vector in a given application? Monte Carlo cross-validation (Burman, 1998) has given the traditional way to proceed. The work of Hyvärinen et al. (2001) on independent component analysis (ICA) looks promising for electrocardiogram data. On addition, a package in R to perform ICA is available.

Appendix A: R codes

A.1 Original Electrocardiogram Data

```
# Plot the original electrocardiogram data for volunteers and
patients # Read the data

tangp1<-read.table('p1.txt')
tangp2<-read.table('p2.txt')
tangp3<-read.table('p3.txt')
tangv1<-read.table('v1.txt')
tangv2<-read.table('v2.txt')
tangv3<-read.table('v3.txt')
tv1<-tangv1$V2 tv2<-tangv2$V2
tv3<-tangv3$V2 tp1<-tangp1$V2
tp2<-tangp2$V2 tp3<-tangp3$V2

# Lengths of electrocardiogram data for volunteers and patients

nv1<-length(tv1)
nv2<-length(tv2)
nv3<-length(tv3)
np1<-length(tp1)
np2<-length(tp2)
np3<-length(tp3)
ttv1<-tv1[1:10000]
ttv2<-tv2[1:10000]
ttv3<-tv3[1:10000]
ttp1<-tp1[1:10000]
ttp2<-tp2[1:10000]
ttp3<-tp3[1:10000]

# Plot the electrocardiogram data of volunteers

windows()
par(mfrow<-c(3,1))
ystand<-seq(-0.7,1.8,2.5/10000)
plot(ystand,ylab='volunteer 1',col='white')
lines(ttv1,type='l')
plot(ystand,ylab='volunteer 2',col='white')
lines(ttv2,type='l')
plot(ystand,ylab='volunteer 3',col='white')
```

```
lines(ttv3,type='l')

# Plot the electrocardiogram data of patients

windows()
par(mfrow<-c(3,1))
ystand<-seq(-0.7,1.8,2.5/10000)
plot(ystand,ylab='patient 1',col='white')
lines(ttp1,type='l')
plot(ystand,ylab='patient 2',col='white')
lines(ttp2,type='l')
plot(ystand,ylab='patient 3',col='white')
lines(ttp3,type='l')
```

A.2 Segment and extracting statistics from the electrocardiogram data for volunteers and patients

```
# Function of finding the position of the minimum of a vector

minposition<-function(y)
 {
  l<-length(y)
  minpos<-1
  minvalue<-y[1]
  for(i in 2:l)
   {
    if((y[i]<minvalue)==TRUE)
     {
      minvalue<-y[i]
      minpos<-i
     }
   }
  minpos
 }

# Function of finding the position of the maximum of a vector

maxposition<-function(y)
 {
   l<-length(y)
   maxpos<-1
   maxvalue<-y[1]
   for(i in 2:l)
```

```
    {
     if((y[i]>maxvalue)==TRUE)
      {
        maxvalue<-y[i]
        maxpos<-i
      }
    }
   maxpos
 }

# Segment and statistics of the electrocardiogram data for patient1
# Read the original data of patient 1

tangp1<-read.table('p1.txt')
ttv1<-tangp1$V2
n<-length(tv1)
ta<-0.3

# Get the segment position vector for patient 1

tpos<-array(0,n)
i<-1
flagb<-0
flagn<-0
for (j in 2:n)
 {
   if ((ttv1[j]>ta)==TRUE)
     {
       flagb<-flagn
       flagn<-1
     }
   if((ttv1[j]<ta)==TRUE)
     {
       flagb<-flagn
       flagn<-0
     }
   if(flagb 6= flagn)
     {
       tpos[i]<-j
       i<-i+1
     }
 }
m<-1
```

```
for (j in 1:n)
 {
  if((tpos[j]==0)==FALSE)
   {
     m<-m+1
   }
 }
tposfinal<-array(0,m-1)
i<-1
for (j in 1:n)
  {
    if((tpos[j]==0)==FALSE)
      {
        tposfinal[i]<-tpos[j]
        i<-i+1
      }
  }
l<-length(tposfinal)
tposmax<-array(0,l/2)
tposfinall<-array(0,l)
j<-1 for (i in 1:l)
  {
    if (i%%2==1)
      {
        tempv<-ttv1[(tposfinal[i]-50):tposfinal[i]]
        minpos<-minposition(tempv)
        tposfinall[i]<-tposfinal[i]-50+minpos
      }
    if (i%%2==0)
      {
        tempv<-ttv1[tposfinal[i]:(tposfinal[i]+50)]
        k<-i-1
        tempv1<-ttv1[tposfinal[k]:tposfinal[i]]
        minpos<-minposition(tempv)
        maxpos<-maxposition(tempv1)
        tposfinall[i]<-tposfinal[i]+minpos
        tposmax[j]<-tposfinal[k]+maxpos-1
        j<-j+1
      }
   }

# Plot the segment of electrocardiogram data for patient 1
```

```
plot(ttv1[1:3500],ylab='value of Patient 1',xlab='Time',type='l')
points(tposfinall[1],ttv1[tposfinall[1]],pch=23)
text(tposfinall[1],ttv1[tposfinall[1]],label='cp',pos=2)
points(tposmax[1],ttv1[tposmax[1]],pch=23,col='red')
text(tposmax[1],ttv1[tposmax[1]],label='maxp',pos=4)
points(tposfinall[2],ttv1[tposfinall[2]],pch=23)
text(tposfinall[2],ttv1[tposfinall[2]],label='cp',pos=4)
points(tposfinall[3],ttv1[tposfinall[3]],pch=23)
text(tposfinall[3],ttv1[tposfinall[3]],label='cp',pos=2)
points(tposmax[2],ttv1[tposmax[2]],pch=23,col='red')
text(tposmax[2],ttv1[tposmax[2]],label='maxp',pos=4)
points(tposfinall[4],ttv1[tposfinall[4]],pch=23)
text(tposfinall[4],ttv1[tposfinall[4]],label='cp',pos=4)
points(tposfinall[5],ttv1[tposfinall[5]],pch=23)
text(tposfinall[5],ttv1[tposfinall[5]],label='cp',pos=2)
points(tposmax[3],ttv1[tposmax[3]],pch=23,col='red')
text(tposmax[3],ttv1[tposmax[3]],label='maxp',pos=4)
points(tposfinall[6],ttv1[tposfinall[6]],pch=23)
text(tposfinall[6],ttv1[tposfinall[6]],label='cp',pos=4)
points(tposfinall[7],ttv1[tposfinall[7]],pch=23)
text(tposfinall[7],ttv1[tposfinall[7]],label='cp',pos=2)
points(tposmax[4],ttv1[tposmax[4]],pch=23,col='red')
text(tposmax[4],ttv1[tposmax[4]],label='maxp',pos=4)
points(tposfinall[8],ttv1[tposfinall[8]],pch=23)
text(tposfinall[8],ttv1[tposfinall[8]],label='cp',pos=4)

# Plot the higher peak and lower peak of electrocardiogram data
patient 1, separately

window() par(mfrow<-c(1,2))
plot(ttv1[tposfinall[1]:tposfinall[2]],
xlab='Time',type='l')
plot(ttv1[tposfinall[2]:tposfinall[3]],
xlab='Time',type='l')

# Get the variation vector of the position of maximum of patient 1

l<-length(tposmax)
tposmaxa<-tposmax[1:(l-1)]
tposmaxb<-tposmax[2:l]
tposmaxv<-array(0,(l-1))
tposmaxv<-tposmaxb-tposmaxa
tposmaxvp1<-tposmaxv
```

```
# Get the mean of position interval, maximum, range, mean of
interval for patient 1

tposfinal<-tposfinall
leng<-length(tposfinal)
posinterval<-array(0,leng/2)
posintervalL<-array(0,leng/2-1)

for(j in 1:(leng/2))
   {
     posinterval[j]<-tposfinal[j*2]-tposfinal[j*2-1]
   }
for(j in 1:(leng/2-1))
   {
     posintervalL[j]<-tposfinal[j*2+1]-tposfinal[j*2]
   }

maximum<-array(0,leng/2) meaninterval<-array(0,leng/2-1)
minimum<-array(0,leng/2-1) maximumLow<-array(0,leng/2-1)

for (j in 1:(leng/2))
  {
    a<-tposfinal[2*j-1]
    b<-tposfinal[2*j]
    maximum[j]<-max(ttv1[a:b])
  }
for (j in 1:(leng/2-1))
  {
    test<-j
    minimum[j]<-min(ttv1[tposfinal[2*test]:tposfinal[2*test+1]])
    st<-tposfinal[2*test]+10
    en<-tposfinal[2*test+1]-10
    maximumLow[j]<-max(ttv1[st:en])
    meaninterval[j]<-mean(ttv1[tposfinal[2*test]:tposfinal[2*test+1]])
  }

meanmaximum<-mean(maximum)
meanminimum<-mean(minimum)
meanmaximumLow<-mean(maximumLow)
rangeHigh<-meanmaximum-meanminimum
rangeLow<-meanmaximumLow-meanminimum
meanposinterval<-mean(posinterval)/1000
```

```
meanposintervalL<-mean(posintervalL)/1000
meanlowinterval<-mean(meaninterval)
mrangeH<-maximum[1:(length(maximum)-1)]-minimum
mrangeL<-maximumLow[1:(length(maximum)-1)]-minimum

# Get the feature vector for the electrocardiogram data for patient
1

result<-c(meanmaximum,rangeHigh,meanmaximumLow,rangeLow,meanposinterval,
meanposintervalL,meanlowinterval)

# Get the vectors of maximum value of higher peak, range of higher
peak, maximum value of lower peak, range of lower peak, width of
higher peak, width of lower peak and average value of lower peak for
the electrocardiogram data

maximump1<-maximum
mrangeHp1<-mrangeH
maximumLowp1<-maximumLow
mrangeLp1<-mrangeL
posintervalp1<-posinterval
posintervalLp1<-posintervalL
meanintervalp1<-meaninterval

A.3 Comparing Boxplots, densities of the electrocardiogram data for
group of patients and group of volunteers

# Comparing boxplots and densities of maximum value of higher peak
of volunteers and patients

n1<-length(maximumv1)
n2<-length(maximumv2)
n3<-length(maximumv3)
n4<-length(maximump1)
n5<-length(maximump2)
n6<-length(maximump3)
boxf<-c(maximumv1,maximump1,maximumv2,maximump2,maximumv3,maximump3)
c1<-array('v1',n1)
c2<-array('v2',n2)
c3<-array('v3',n3)
c4<-array('p1',n4)
c5<-array('p2',n5)
c6<-array('p3',n6)
```

```
cf<-c(c1,c2,c3,c4,c5,c6)
cf<-factor(cf)
boxplot(boxf_cf)
plot(density(maximumv1),xlim<-c(0,2.5),ylim<-c(0,15),type='l',lty=2)
lines(density(maximumv2),type='l',lty=2,col='blue')
lines(density(maximumv3),type='l',lty=2,ol='red')
lines(density(maximump1),type='l')
lines(density(maximump2),type='l',col='blue')
lines(density(maximump3),type='l',col='red')

# Comparing boxplots and densities of range of higher peak of
volunteers and patients

n1<-length(mrangeHv1)
n2<-length(mrangeHv2)
n3<-length(mrangeHv3)
n4<-length(mrangeHp1)
n5<-length(mrangeHp2)
n6<-length(mrangeHp3)
boxf<-c(mrangeHv1,mrangeHv2,mrangeHv3,mrangeHp1,mrangeHp2,mrangeHp3)
c1<-array('v1',n1)
c2<-array('v2',n2)
c3<-array('v3',n3)
c4<-array('p1',n4)
c5<-array('p2',n5)
c6<-array('p3',n6)
cf<-c(c1,c2,c3,c4,c5,c6)
cf<-factor(cf)
boxplot(boxf_cf)
plot(density(mrangeHv1),xlim<-c(0,2.5),ylim=c(0,18),type='l',lty=2)
lines(density(mrangeHv2),type='l',lty=2,col='blue')
lines(density(mrangeHv3),type='l',lty=2,col='red')
lines(density(mrangeHp1),type='l')
lines(density(mrangeHp2),type='l',col='blue')
lines(density(mrangeHp3),type='l',col='red')

# Comparing boxplots and densities of maximum value of lower peak of
volunteers and patients

n1<-length(maximumLowv1)
n2<-length(maximumLowv2)
n3<-length(maximumLowv3)
n4<-length(maximumLowp1)
```

```
n5<-length(maximumLowp2)
n6<-length(maximumLowp3)
boxf<-c(maximumLowv1,maximumLowv2,maximumLowv3,maximumLowp1,
maximumLowp2,maximumLowp3)
c1<-array('v1',n1)
c2<-array('v2',n2)
c3<-array('v3',n3)
c4<-array('p1',n4)
c5<-array('p2',n5)
c6<-array('p3',n6)
cf<-c(c1,c2,c3,c4,c5,c6)
cf<-factor(cf)
boxplot(boxf_cf)
plot(density(maximumLowv1),xlim=c(-0.15,0.85),ylim=c(0,18),type='l',lty=2)
lines(density(maximumLowv2),type='l',lty=2,col='blue')
lines(density(maximumLowv3),type='l',lty=2,col='red')
lines(density(maximumLowp1),type='l')
lines(density(maximumLowp2),type='l',col='blue')
lines(density(maximumLowp3),type='l',col='red')

# Comparing boxplots and densities of range of lower peak of
 volunteers and patients

n1<-length(mrangeLv1)
n2<-length(mrangeLv2)
n3<-length(mrangeLv3)
n4<-length(mrangeLp1)
n5<-length(mrangeLp2)
n6<-length(mrangeLp3)
boxf<-c(mrangeLv1,mrangeLv2,mrangeLv3,mrangeLp1,mrangeLp2,mrangeLp3)
c1<-array('v1',n1)
c2<-array('v2',n2)
c3<-array('v3',n3)
c4<-array('p1',n4)
c5<-array('p2',n5)
c6<-array('p3',n6)
cf<-c(c1,c2,c3,c4,c5,c6)
cf<-factor(cf)
boxplot(boxf_cf)
plot(density(mrangeLv1),xlim=c(0,0.85),ylim=c(0,22),type='l',lty=2)
lines(density(mrangeLv2),type='l',lty=2,col='blue')
lines(density(mrangeLv3),type='l',lty=2,col='red')
lines(density(mrangeLp1),type='l')
```

```
lines(density(mrangeLp2),type='l',col='blue')
lines(density(mrangeLp3),type='l',col='red')

# Comparing boxplots and densities of width of higher peak of
volunteers and patients

n1<-length(posintervalv1)
n2<-length(posintervalv2)
n3<-length(posintervalv3)
n4<-length(posintervalp1)
n5<-length(posintervalp2)
n6<-length(posintervalp3)
boxf<-c(posintervalv1,posintervalv2,posintervalv3,posintervalp1,
posintervalp2,posintervalp3)
c1<-array('v1',n1)
c2<-array('v2',n2)
c3<-array('v3',n3)
c4<-array('p1',n4)
c5<-array('p2',n5)
c6<-array('p3',n6)
cf<-c(c1,c2,c3,c4,c5,c6)
cf<-factor(cf)
boxplot(boxf_cf)
plot(density(posintervalv1),xlim=c(0,100),ylim=c(0,0.4),type='l',lty=2)
lines(density(posintervalv2),type='l',lty=2,col='blue')
lines(density(posintervalv3),type='l',lty=2,col='red')
lines(density(posintervalp1),type='l')
lines(density(posintervalp2),type='l',col='blue')
lines(density(posintervalp3),type='l',col='red')

# Comparing boxplots and densities of width of lower peak of
 volunteers and patients

n1<-length(posintervalLv1)
n2<-length(posintervalLv2)
n3<-length(posintervalLv3)
n4<-length(posintervalLp1)
n5<-length(posintervalLp2)
n6<-length(posintervalLp3)
boxf<-c(posintervalLv1,posintervalLv2,posintervalLv3,posintervalLp1,
posintervalLp2,posintervalLp3)
c1<-array('v1',n1)
c2<-array('v2',n2)
```

```
c3<-array('v3',n3)
c4<-array('p1',n4)
c5<-array('p2',n5)
c6<-array('p3',n6)
cf<-c(c1,c2,c3,c4,c5,c6)
cf<-factor(cf)
boxplot(boxf_cf)
plot(density(posintervalLv1),xlim=c(0,1500),ylim=c(0,0.03),type='l',lty=2)
lines(density(posintervalLv2),type='l',lty=2,col='blue')
lines(density(posintervalLv3),type='l',lty=2,col='red')
lines(density(posintervalLp1),type='l')
lines(density(posintervalLp2),type='l',col='blue')
lines(density(posintervalLp3),type='l',col='red')

# Comparing boxplots and densities of average value of lower
 peak of volunteers and patients

n1<-length(meanintervalv1)
n2<-length(meanintervalv2)
n3<-length(meanintervalv3)
n4<-length(meanintervalp1)
n5<-length(meanintervalp2)
n6<-length(meanintervalp3)
boxf<-c(meanintervalv1,meanintervalv2,meanintervalv3,meanintervalp1,
meanintervalp2,meanintervalp3)
c1<-array('v1',n1)
c2<-array('v2',n2)
c3<-array('v3',n3)
c4<-array('p1',n4)
c5<-array('p2',n5)
c6<-array('p3',n6)
cf<-c(c1,c2,c3,c4,c5,c6)
cf<-factor(cf)
boxplot(boxf_cf)
plot(density(meanintervalv1),xlim=c(-0.4,0.6),ylim=c(0,120),type='l',lty=2)
lines(density(meanintervalv2),type='l',lty=2,col='blue')
lines(density(meanintervalv3),type='l',lty=2,col='red')
lines(density(meanintervalp1),type='l')
lines(density(meanintervalp2),type='l',col='blue')
lines(density(meanintervalp3),type='l',col='red')

# Comparing boxplots and densities of position variation vector of
 maximum of volunteers and patients
```

```
n1<-length(tposmaxvv1)
n2<-length(tposmaxvv2)
n3<-length(tposmaxvv3)
n4<-length(tposmaxvp1)
n5<-length(tposmaxvp2)
n6<-length(tposmaxvp3)
boxf<-c(tposmaxvv1,tposmaxvv2,tposmaxvv3,tposmaxvp1,tposmaxvp2,tposmaxvp3)
c1<-array('v1',n1)
c2<-array('v2',n2)
c3<-array('v3',n3)
c4<-array('p1',n4)
c5<-array('p2',n5)
c6<-array('p3',n6)
cf<-c(c1,c2,c3,c4,c5,c6)
cf<-factor(cf) boxplot(boxf_cf)
plot(density(tposmaxvv1),xlim=c(0,1200),ylim=c(0,0.03),type='l',lty=2)
lines(density(tposmaxvv2),type='l',lty=2,col='blue')
lines(density(tposmaxvv3),type='l',lty=2,col='red')
lines(density(tposmaxvp1),type='l')
lines(density(tposmaxvp2),type='l',col='blue')
lines(density(tposmaxvp3),type='l',col='red')
```

A.4 Comparing single statistics of the electrocardiogram data for group of patients and group of volunteers

```
# Read the file of feature vectors for the electrocardiogram data

ttresult1<-read.table('result.txt')
ttmaximumH<-ttresult1$V1
ttrangeH<-ttresult1$V2
ttmaximumL<-ttresult1$V3
ttrangeL<-ttresult1$V4
ttintervalH<-ttresult1$V5
ttintervalL<-ttresult1$V6
ttvalueintervalL<-ttresult1$V7

# Compare maximum of higher peak and lower peak of volunteers and
patients

windows() par(mfrow<-c(1,2)) y<-c(0,1,2) plot(y,type='l',ylab='mean
maximum of higher peak') lines(ttmaximumH[1:3],type='l')
```

```
points(1,ttmaximumH[1],pch=24,cex=1)
text(1,ttmaximumH[1],labels='V1',pos=3,cex=0.7)
points(2,ttmaximumH[2],pch=24,cex=1)
text(2,ttmaximumH[2],labels='V2',pos=3,cex=0.7)
points(3,ttmaximumH[3],pch=24,cex=1)
text(3,ttmaximumH[3],labels='V3',pos=3,cex=0.7)
lines(ttmaximumH[4:6],type='l',col='red')
points(1,ttmaximumH[4],cex=1,col='red')
text(1,ttmaximumH[4],labels='P1',pos=1,cex=0.7)
points(2,ttmaximumH[5],cex=1,col='red')
text(2,ttmaximumH[5],labels='P2',pos=1,cex=0.7)
points(3,ttmaximumH[6],cex=1,col='red')
text(3,ttmaximumH[6],labels='P3',pos=1,cex=0.7)
y<-c(-1,0,1)
plot(y,type='l',ylab='mean maximum of lower peak',col='white')
points(2,ttmaximumL[1],pch=24,cex=1)
text(2,ttmaximumL[1],labels='V1',pos=1,cex=0.7)
points(1,ttmaximumL[2],pch=24,cex=1)
text(1,ttmaximumL[2],labels='V2',pos=1,cex=0.7)
points(3,ttmaximumL[3],pch=24,cex=1)
text(3,ttmaximumL[3],labels='V3',pos=1,cex=0.7)
lines(c(ttmaximumL[2],ttmaximumL[1],ttmaximumL[3]),type='l')
lines(ttmaximumL[4:6],type='l',col='red')
points(1,ttmaximumL[4],cex=1,col='red')
text(1,ttmaximumL[4],labels='P1',pos=3,cex=0.7)
points(2,ttmaximumL[5],cex=1,col='red')
text(2,ttmaximumL[5],labels='P2',pos=3,cex=0.7)
points(3,ttmaximumL[6],cex=1,col='red')
text(3,ttmaximumL[6],labels='P3',pos=3,cex=0.7)


# Compare range of higher peak and lower peak, and the ratio of them
for volunteers and patients

windows()
par(mfrow<-c(1,3))
y<-c(0,1,2)
plot(y,type='l',ylab='mean
range of higher peak ',col='white')
lines(ttrangeH[1:3],type='l')
points(1,ttrangeH[1],pch=24,cex=1)
text(1,ttrangeH[1],labels='V1',cex=0.7,pos=3)
points(2,ttrangeH[2],pch=24,cex=1)
```

```
text(2,ttrangeH[2],labels='V2',cex=0.7,pos=3)
points(3,ttrangeH[3],pch=24,cex=1)
text(3,ttrangeH[3],labels='V3',cex=0.7,pos=3)
lines(ttrangeH[4:6],type='l',col='red')
points(1,ttrangeH[4],cex=1,col='red')
text(1,ttrangeH[4],labels='P1',cex=0.7,pos=1)
points(2,ttrangeH[5],cex=1,col='red')
text(2,ttrangeH[5],labels='P2',cex=0.7,pos=1)
points(3,ttrangeH[6],cex=1,col='red')
text(3,ttrangeH[6],labels='P3',cex=0.7,pos=1)
plot(y,type='l',ylab='mean range of lower peak',col='white')
lines(ttrangeL[1:3],type='l')
points(1,ttrangeL[1],pch=24,cex=1)
text(1,ttrangeL[1],labels='V1',cex=0.7,pos=3)
points(2,ttrangeL[2],pch=24,cex=1)
text(2,ttrangeL[2],labels='V1',cex=0.7,pos=3)
points(3,ttrangeL[3],pch=24,cex=1)
text(3,ttrangeL[3],labels='V3',cex=0.7,pos=1)
lines(ttrangeL[4:6],type='l',col='red')
points(1,ttrangeL[4],cex=1,col='red')
text(1,ttrangeL[4],labels='P1',cex=0.7,pos=1)
points(2,ttrangeL[5],cex=1,col='red')
text(2,ttrangeL[5],labels='P2',cex=0.7,pos=1)
points(3,ttrangeL[6],cex=1,col='red')
text(3,ttrangeL[6],labels='P3',cex=0.7,pos=3) y<-c(1,2,3,4)
rangeratio<-array(0,6)
for(j in 1:6)
 {
  rangeratio[j]<-ttrangeH[j]/ttrangeL[j]
 }
plot(y,type='l',ylab='mean
Rangeratio:RangeHihgerpeak/RangLowerpeak',col='white')
lines(rangeratio[1:3],type='l')
points(1,rangeratio[1],pch=24,cex=1)
text(1,rangeratio[1],labels='V1',cex=0.7,pos=3)
points(2,rangeratio[2],pch=24,cex=1)
text(2,rangeratio[2],labels='V2',cex=0.7,pos=3)
points(3,rangeratio[3],pch=24,cex=1)
text(3,rangeratio[3],labels='V3',cex=0.7,pos=3)
lines(rangeratio[4:6],type='l',col='red')
points(1,rangeratio[4],cex=1,col='red')
text(1,rangeratio[4],labels='P1',cex=0.7,pos=1)
points(2,rangeratio[5],cex=1,col='red')
```

```
text(2,rangeratio[5],labels='P2',cex=0.7,pos=1)
points(3,rangeratio[6],cex=1,col='red')
text(3,rangeratio[6],labels='P3',cex=0.7,pos=1)

# Compare width of higher peak and lower peak, and sum of them for
volunteers and patients

windows()
par(mfrow<-c(1,3))
y<-c(0,0.1,0.2)
plot(y,type='l',ylab='mean width of higher peak',col='white')
lines(ttintervalH[1:3],type='l')
points(1,ttintervalH[1],pch=24,cex=1)
text(1,ttintervalH[1],labels='V1',cex=0.7,pos=1)
points(2,ttintervalH[2],pch=24,cex=1)
text(2,ttintervalH[2],labels='V2',cex=0.7,pos=3)
points(3,ttintervalH[3],pch=24,cex=1)
text(3,ttintervalH[3],labels='V3',cex=0.7,pos=3)
lines(ttintervalH[4:6],type='l',col='red')
points(1,ttintervalH[4],cex=1,col='red')
text(1,ttintervalH[4],labels='P1',cex=0.7,pos=3)
points(2,ttintervalH[5],cex=1,col='red')
text(2,ttintervalH[5],labels='P2',cex=0.7,pos=1)
points(3,ttintervalH[6],cex=1,col='red')
text(3,ttintervalH[6],labels='P3',cex=0.7,pos=1) y<-c(0,0.5,1)
plot(y,type='l',ylab='mean width of lower peak',col='white')
lines(ttintervalL[1:3],type='l')
points(1,ttintervalL[1],pch=24,cex=1)
text(1,ttintervalL[1],labels='V1',cex=0.7,pos=1)
points(2,ttintervalL[2],pch=24,cex=1)
text(2,ttintervalL[2],labels='V2',cex=0.7,pos=3)
points(3,ttintervalL[3],pch=24,cex=1)
text(3,ttintervalL[3],labels='V3',cex=0.7,pos=3)
lines(ttintervalL[4:6],type='l',col='red')
points(1,ttintervalL[4],cex=1,col='red')
text(1,ttintervalL[4],labels='P1',cex=0.7,pos=3)
points(2,ttintervalL[5],cex=1,col='red')
text(2,ttintervalL[5],labels='P2',cex=0.7,pos=1)
points(3,ttintervalL[6],cex=1,col='red')
text(3,ttintervalL[6],labels='P3',cex=0.7,pos=1)
RRinterval<-array(0,6) for(j in 1:6)
 {
   RRinterval[j]<-ttintervalH[j]+ttintervalL[j]
```

```
 }
y<-c(0,1,2)
plot(y,type='l',ylab='mean width of higer and lower
peak',col='white')
lines(RRinterval[1:3],type='l')
points(1,RRinterval[1],pch=24,cex=1)
text(1,RRinterval[1],labels='V1',cex=0.7,pos=1)
points(2,RRinterval[2],pch=24,cex=1)
text(2,RRinterval[2],labels='V2',cex=0.7,pos=3)
points(3,RRinterval[3],pch=24,cex=1)
text(3,RRinterval[3],labels='V3',cex=0.7,pos=3)
lines(RRinterval[4:6],type='l',col='red')
points(1,RRinterval[4],cex=1,col='red')
text(1,RRinterval[4],labels='P1',cex=0.7,pos=3)
points(2,RRinterval[5],cex=1,col='red')
text(2,RRinterval[5],labels='P2',cex=0.7,pos=1)
points(3,RRinterval[6],cex=1,col='red')
text(3,RRinterval[6],labels='P3',cex=0.7,pos=1)

# Compare the average value of lower peak for volunteers and
patients

y<-c(0,0.25,0.5)
absttvalueintervalL<-abs(ttvalueintervalL)
plot(y,type='l',ylab='mean value of lower peak',col='white')
points(2,absttvalueintervalL[1],pch=24,cex=1)
text(2,absttvalueintervalL[1],labels='V1',cex=0.7,pos=3)
points(1,absttvalueintervalL[2],pch=24,cex=1)
text(1,absttvalueintervalL[2],labels='V2',cex=0.7,pos=3)
points(3,absttvalueintervalL[3],pch=24,cex=1)
text(3,absttvalueintervalL[3],labels='V3',cex=0.7,pos=3)
lines(c(absttvalueintervalL[2],absttvalueintervalL[1],
absttvalueintervalL[3]),type='l')
lines(absttvalueintervalL[4:6],type='l',col='red')
points(1,absttvalueintervalL[4],cex=1,col='red')
text(1,absttvalueintervalL[4],labels='P1',cex=0.7,pos=1)
points(2,absttvalueintervalL[5],cex=1,col='red')
text(2,absttvalueintervalL[5],labels='P2',cex=0.7,pos=1)
points(3,absttvalueintervalL[6],cex=1,col='red')
text(3,absttvalueintervalL[6],labels='P3',cex=0.7,pos=1)
```

A.5 Clustering analysis of the feature vectors for the

```
electrocardiogram data for group of patients and group of volunteers

# Distance function for two vectors

distance<-function(y1,y2)
 {
   n<-length(y1)
   sum<-0
   for(i in 1:n)
    {
      sum<-sum+(y1[i]-y2[i])*(y1[i]-y2[i])
    }
   dis<-sqrt(sum)
   dis
 }


# K-means clustering analysis for the electrocardiogram data of
volunteers and patients # Read the file of the feature vectors for
volunteers and patients

ttresult1<-read.table('result.txt')
ttmaximumH<-ttresult1$V1
ttrangeH<-ttresult1$V2
ttmaximumL<-ttresult1$V3
ttrangeL<-ttresult1$V4
ttintervalH<-ttresult1$V5
ttintervalL<-ttresult1$V6
ttvalueintervalL<-ttresult1$V7
v1<-c(ttmaximumH[1],ttrangeH[1],ttmaximumL[1],ttrangeL[1],ttintervalH[1],
ttintervalL[1],ttvalueintervalL[1])
v2<-c(ttmaximumH[2],ttrangeH[2],ttmaximumL[2],ttrangeL[2],ttintervalH[2],
ttintervalL[2],ttvalueintervalL[2])
v3<-c(ttmaximumH[3],ttrangeH[3],ttmaximumL[3],ttrangeL[3],ttintervalH[3],
ttintervalL[3],ttvalueintervalL[3])
p1<-c(ttmaximumH[4],ttrangeH[4],ttmaximumL[4],ttrangeL[4],ttintervalH[4],
ttintervalL[4],ttvalueintervalL[4])
p2<-c(ttmaximumH[5],ttrangeH[5],ttmaximumL[5],ttrangeL[5],ttintervalH[5],
ttintervalL[5],ttvalueintervalL[5])
p3<-c(ttmaximumH[6],ttrangeH[6],ttmaximumL[6],ttrangeL[6],ttintervalH[6],
ttintervalL[6],ttvalueintervalL[6])
tvp<-c(v1,v2,v3,p1,p2,p3)
center1<-c(0.5,0.9,0.1,0.6,0.03,0.7,-0.1)
```

```
center2<-c(0.2,0.6,0.2,0.5,0.01,0.9,0)
clusterpos<-c(0,0,0,0,0,0)
resultclusterpos<-clusterpos
resultcenter1<-center1
resultcenter2<-center2
stopdistance<-0.01
tempdistance<-1

while(tempdistance>stopdistance)
 {
  # new clustering
  for (i in 1:6)
  {
    s<-7*(i-1)+1
    tempv<-tvp[s:(s+6)]
    d1<-distance(tempv,center1)
    d2<-distance(tempv,center2)
    if((d1<-d2)==TRUE)
     {
       clusterpos[i]<-1
     }
    if((d1>d2)==TRUE)
     {
       clusterpos[i]<-2
     }
  }
  # new centers and tempdistance
  c1num<-0
  c2num<-0
  c1new<-c(0,0,0,0,0,0,0)
  c2new<-c(0,0,0,0,0,0,0)
  for(i in 1:6)
   {
    s<-7*(i-1)+1
    tempv<-tvp[s:(s+6)]
    if((clusterpos[i]==1)==TRUE)
     {
       c1num<-c1num+1
       c1new<-c1new+tempv
     }
    if((clusterpos[i]==2)==TRUE)
     {
       c2num<-c2num+1
```

```
      c2new<-c2new+tempv
    }
  }
  c1new<-c1new/c1num
  c2new<-c2new/c2num
  tempdistance<-distance(center1,c1new)+distance(center2,c2new)
  resultcenter1<-c(resultcenter1,'#',c1new)
  resultcenter2<-c(resultcenter2,'#',c2new)
  resultclusterpos<-c(resultclusterpos,'#',clusterpos)
  center1<-c1new
  center2<-c2new
}


# Plot the result of K-means clustering method of volunteers and
patients

center<-c(0,2)
index<-c(0,1)
plot(center,index,ylab='Mean value of
range of higher peak',col='white',xlab='Mean value of range of lower
peak')
points(ttrangeH[1],ttrangeL[1],pch=24)
text(x=ttrangeH[1],y=ttrangeL[1],labels='V1',pos=4)
points(ttrangeH[2],ttrangeL[2],pch=24)
text(x=ttrangeH[2],y=ttrangeL[2],labels='V2',pos=4)
points(ttrangeH[3],ttrangeL[3],pch=24)
text(x=ttrangeH[3],y=ttrangeL[3],labels='V3',pos=4)
points(ttrangeH[4],ttrangeL[4],col='red')
text(x=ttrangeH[4],y=ttrangeL[4],labels='P1',pos=2)
points(ttrangeH[5],ttrangeL[5],col='red')
text(x=ttrangeH[5],y=ttrangeL[5],labels='P2',pos=2)
points(ttrangeH[6],ttrangeL[6],col='red')
text(x=ttrangeH[6],y=ttrangeL[6],labels='P3',pos=2)
points(0.9,0.6,pch=23) text(0.9,0.6,labels='C10',pos=1,cex=0.7)
points(1.4712,0.6147,pch=23)
text(1.4712,0.6147,labels='C11',pos=1,cex=0.7)
points(1.6086,0.6185,pch=23)
text(1.6086,0.6185,labels='C12',pos=1,cex=0.7)
points(0.6,0.5,pch=22,col='red')
text(0.6,0.5,labels='C20',pos=1,cex=0.7)
points(0.6675,0.5313,pch=22,col='red')
text(0.6675,0.5313,labels='C21',pos=1,cex=0.7)
```

```
points(0.7945,0.5653,pch=22,col='red')
text(0.7945,0.5653,labels='C22',pos=1,cex=0.7)
arrows(0.9,0.6,1.4712,0.6147,type='l')
arrows(1.4712,0.6147,1.6086,0.6185,type='l')
arrows(0.6,0.5,0.6675,0.5313,tpye='l',col='red')
arrows(0.6675,0.5313,0.7945,0.5653,tpye='l',col='red')
arrows(1.3,0,0.9,1.1,type='l')
text(x=1.2,y=0.3,labels='Final
Boundary',pos=4)

# Hierarchical clustering analysis for the electrocardiogram data of
volunteers and patients # Read the file of the feature vectors for
volunteers and patients

ttresult1<-read.table('result.txt')
ttmaximumH<-ttresult1$V1
ttrangeH<-ttresult1$V2
ttmaximumL<-ttresult1$V3
ttrangeL<-ttresult1$V4
ttintervalH<-ttresult1$V5
ttintervalL<-ttresult1$V6
ttvalueintervalL<-ttresult1$V7
v1<-c(ttmaximumH[1],ttrangeH[1],ttmaximumL[1],ttrangeL[1],ttintervalH[1],
ttintervalL[1],ttvalueintervalL[1])
v2<-c(ttmaximumH[2],ttrangeH[2],ttmaximumL[2],ttrangeL[2],ttintervalH[2],
ttintervalL[2],ttvalueintervalL[2])
v3<-c(ttmaximumH[3],ttrangeH[3],ttmaximumL[3],ttrangeL[3],ttintervalH[3],
ttintervalL[3],ttvalueintervalL[3])
p1<-c(ttmaximumH[4],ttrangeH[4],ttmaximumL[4],ttrangeL[4],ttintervalH[4],
ttintervalL[4],ttvalueintervalL[4])
p2<-c(ttmaximumH[5],ttrangeH[5],ttmaximumL[5],ttrangeL[5],ttintervalH[5],
ttintervalL[5],ttvalueintervalL[5])
p3<-c(ttmaximumH[6],ttrangeH[6],ttmaximumL[6],ttrangeL[6],ttintervalH[6],
ttintervalL[6],ttvalueintervalL[6])
tvp<-c(v1,v2,v3,p1,p2,p3)
resultmergeorder<-c(0,0)
resultcenter<-tvp

for(i in 1:4)
 {
   com<-(7-i)*(6-i)/2
   tempdistance<-array(100,com)
   tempdistancex<-array(0,com)
```

```
   tempdistancey<-array(0,com)
   j<-1
   for(m in 1:(6-i))
    {
      s1<-7*(m-1)+1
      tempv1<-tvp[s1:(s1+4)]
      for(n in (m+1):(7-i))
       {
         s2<-7*(n-1)+1
         tempv2<-tvp[s2:(s2+4)]
         tempdistance[j]<-distance(tempv1,tempv2)
         tempdistancex[j]<-m
         tempdistancey[j]<-n
         j<-j+1
       }
  }
temppos<-minpostion(tempdistance)
resultmergeorder<-c(resultmergeorder,'#',c(tempdistancex[temppos],
tempdistancey[temppos]))
newleng<-7*(6-i)
newtvp<-array(0,newleng)
new<-1
for (k in 1:(7-i))
  {
     if(k!<-tempdistancex[temppos]& k!<-tempdistancey[temppos])
       {
         for (t in 1:7)
           {
             m<-7*(new-1)+t
             n<-7*(k-1)+t
             newtvp[m]<-tvp[n]}
             new<-new+1
           }
       }
     m<-tempdistancex[temppos]-1
     n<-tempdistancey[temppos]-1
     newtvp[(newleng-6):newleng]<-(tvp[(7*m+1):(7*(m+1))]+
     tvp[(7*n+1):(7*(n+1))])/2
     tvp<-array(0,newleng)
     tvp<-newtvp
     resultcenter<-c(resultcenter,'#',tvp)
  }
```

```
# Andrew-Plots clustering analysis for the electrocardiogram data of
volunteers and patients

# Read the file of the feature vectors for volunteers and patients

ttresult1<-read.table('result.txt')
ttmaximumH<-ttresult1$V1
ttrangeH<-ttresult1$V2
ttmaximumL<-ttresult1$V3
ttrangeL<-ttresult1$V4
ttintervalH<-ttresult1$V5
ttintervalL<-ttresult1$V6
ttvalueintervalL<-ttresult1$V7
v1<-c(ttmaximumH[1],ttrangeH[1],ttmaximumL[1],ttrangeL[1],ttintervalH[1],
ttintervalL[1],ttvalueintervalL[1])
v2<-c(ttmaximumH[2],ttrangeH[2],ttmaximumL[2],ttrangeL[2],ttintervalH[2],
ttintervalL[2],ttvalueintervalL[2])
v3<-c(ttmaximumH[3],ttrangeH[3],ttmaximumL[3],ttrangeL[3],ttintervalH[3],
ttintervalL[3],ttvalueintervalL[3])
p1<-c(ttmaximumH[4],ttrangeH[4],ttmaximumL[4],ttrangeL[4],ttintervalH[4],
ttintervalL[4],ttvalueintervalL[4])
p2<-c(ttmaximumH[5],ttrangeH[5],ttmaximumL[5],ttrangeL[5],ttintervalH[5],
ttintervalL[5],ttvalueintervalL[5])
p3<-c(ttmaximumH[6],ttrangeH[6],ttmaximumL[6],ttrangeL[6],ttintervalH[6],
ttintervalL[6],ttvalueintervalL[6])

#First-formula Andrew-Plot clustering analysis

window()
AndrewValue<-seq(-1.5,4.5,6/200)
plot(AndrewValue,col='white')
m<-100
y<-array(0,2*m)
t<-(-m:(m-1))*pi/m
y<-v1[1]/sqrt(2)+v1[2]*sin(t)+v1[3]*cos(t)+v1[4]*sin(2*t)+v1[5]*cos(2*t)
+v1[6]*sin(3*t)+v1[7]*cos(3*t) lines(y,type='p')
y<-v2[1]/sqrt(2)+v2[2]*sin(t)+v2[3]*cos(t)+v2[4]*sin(2*t)+v2[5]*cos(2*t)
+v2[6]*sin(3*t)+v2[7]*cos(3*t) lines(y,type='p',col='red')
y<-v3[1]/sqrt(2)+v3[2]*sin(t)+v3[3]*cos(t)+v3[4]*sin(2*t)+v3[5]*cos(2*t)
+v3[6]*sin(3*t)+v3[7]*cos(3*t) lines(y,type='p',col='blue')
y<-p1[1]/sqrt(2)+p1[2]*sin(t)+p1[3]*cos(t)+p1[4]*sin(2*t)+p1[5]*cos(2*t)
+p1[6]*sin(3*t)+p1[7]*cos(3*t) lines(y,type='l')
y<-p2[1]/sqrt(2)+p2[2]*sin(t)+p2[3]*cos(t)+p2[4]*sin(2*t)+p2[5]*cos(2*t)
```

```
+p2[6]*sin(3*t)+p2[7]*cos(3*t) lines(y,type='l',col='red')
y<-p3[1]/sqrt(2)+p3[2]*sin(t)+p3[3]*cos(t)+p3[4]*sin(2*t)+p3[5]*cos(2*t)
+p3[6]*sin(3*t)+p3[7]*cos(3*t) lines(y,type='l',col='blue')

# Second-formula Andrew-Plot clustering analysis

window()
AndrewValue<-seq(-1.5,4.5,6/200)
plot(AndrewValue,col='white')
m<-100
y<-array(0,2*m)
t<-(-m:(m-1))*pi/m
y<-1/sqrt(2)*(v1[1]+v1[2]*(sin(t)+cos(t))+v1[3]*(sin(t)-cos(t))
+v1[4]*(sin(2*t)+cos(2*t))+v1[5]*(sin(2*t)-cos(2*t))
+v1[6]*(sin(3*t)+cos(3*t))+v1[7]*(sin(3*t)-cos(3*t)))
lines(y,type='p')
y<-1/sqrt(2)*(v2[1]+v2[2]*(sin(t)+cos(t))+v2[3]*(sin(t)-cos(t))
+v2[4]*(sin(2*t)+cos(2*t))+v2[5]*(sin(2*t)-cos(2*t))
+v2[6]*(sin(3*t)+cos(3*t))+v2[7]*(sin(3*t)-cos(3*t)))
lines(y,type='p',col='red')
y<-1/sqrt(2)*(v3[1]+v3[2]*(sin(t)+cos(t))+v3[3]*(sin(t)-cos(t))
+v3[4]*(sin(2*t)+cos(2*t))+v3[5]*(sin(2*t)-cos(2*t))
+v2[6]*(sin(3*t)+cos(3*t))+v2[7]*(sin(3*t)-cos(3*t)))
lines(y,type='p',col='blue')
y<-1/sqrt(2)*(p1[1]+p1[2]*(sin(t)+cos(t))+p1[3]*(sin(t)-cos(t))
+p1[4]*(sin(2*t)+cos(2*t))+p1[5]*(sin(2*t)-cos(2*t))
+p1[6]*(sin(3*t)+cos(3*t))+p1[7]*(sin(3*t)-cos(3*t)))
lines(y,type='l')
y<-1/sqrt(2)*(p2[1]+p2[2]*(sin(t)+cos(t))+p2[3]*(sin(t)-cos(t))
+p2[4]*(sin(2*t)+cos(2*t))+p2[5]*(sin(2*t)-cos(2*t))
+p2[6]*(sin(3*t)+cos(3*t))+p2[7]*(sin(3*t)-cos(3*t)))
lines(y,type='l',col='red')
y<-1/sqrt(2)*(p3[1]+p3[2]*(sin(t)+cos(t))+p3[3]*(sin(t)-cos(t))
+p3[4]*(sin(2*t)+cos(2*t))+p3[5]*(sin(2*t)-cos(2*t))
+p3[6]*(sin(3*t)+cos(3*t))+p3[7]*(sin(3*t)-cos(3*t)))
lines(y,type='l',col='blue')
```

Bibliography

Anderson, C. W., Stolz, E.A. and Shamsunder, S. (1998), Multivariate autoregressive models for classification of spontaneous electroencephalographic signals during mental tasks, *IEEE Transactions on Biomedical Engineering*, 45, pp.277-286.

Andrews, D.F. (1972), Plots in high-dimensional data, *Biometrics* 28, pp.125-136.

Becker, D. E. (2006), Fundamentals of electrocardiography interpretation, *Anesthesia Progress.* 53(2), pp.53–64.

Bezdek, J. C. (1981), *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York.

Bradley, P. S. and Fayyad, U. M. (1998), Refining initial points for K-Means clustering, *Microsoft Research Technical Report*, MSR-TR-98-36.

Burman, P. (1989), A comparative study of ordinary cross-validation, v-fold cross-validation, and the repeated learning-testing methods, *Biometrika*, 76, pp.503-514.

Dunn, J. C. (1973), A Fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, *Journal of Cybernetics*, 3, pp.32-57.

Embrechts, P. and Herzberg, A.M. (1991), Variations of Andrew's plots, *International Statistical Review* 59, pp.175-194.

Embrechts, P., Herzberg, A.M., Kalbfleisch, H. K., Traves, W. N., Whitla, J. R. (1995),

An introduction to wavelets with applications to Andrew's plots, *Journal of Computational and Applied Mathematics,* 64, pp.41-56.

Epifanio, I. (2008), Shape descriptors for classification of functional data, *Technometrics*, 50(3), pp.284-234.

Fisher, R. A. (1936), The use of multiple measurements in taxonomic problems, *Annals of Eugenics*, 7(II), pp.179-188.

Hyvärinen, A., Karhunen, J., and Oja, E. (2001), *Independent Component Analysis*, New York: Wiley.

Johnson, S. C. (1967), Hierarchical clustering schemes, *Psychometrika*, 2, pp.241-254.

Khattree, R. and Naik, D. N. (2002), Andrews plots for multivariate data: some new suggestions and applications, *Journal of Statistical Planning and Inference*, 100, pp.411-425.

Khorovets, A. (2000), What is an electrocardiogram (ECG)? *The Internet Journal of Health*, 1(2), http://www.ispub.com/ostia/index.php?xmlFilePath=journals/ijh/vol1n2/ekg.xml.

Lippmann, R. P., Moody, J. and Touretzky, D.S. (1991), *Advances in Neural Information Processing Systems 3*, Morgan Kaufmann, San Mateo, CA.

MacQueen, J. B. (1967), Some methods for classification and analysis of multivariate observations, *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics*

and Probability, Berkeley, University of California Press, 1, pp.281-297.

Ramsay, J. Q., and Silverman, B.W. (2002), *Applied Functional Data Analysis*, New York: Springer.

Ramsay, J. Q., and Silverman, B.W. (2005), *Functional Data Analysis*, New York: Springer.

Wegman, E.J. and Carr, D.B. (1993), Statistical graphics and visualization, *Handbook of Statistics*, Vol. 9., North-Holland, Amsterdam, pp.857-958.

Wegman, E.J., Carr, D.B. and Luo, Q. (1993), Visualizing multivariate data, *Multivariate Analysis, Future Directions,* North-Holland, Amsterdam, pp.423-466.

Wong, R. (2004), Evaluation of heart rate variability and its relationship with cancer related fatigue syndrome in gut, breast and prostate cancer patients, Internal Technical Report, Hamilton Regional Cancer Center, Hamilton, Ontario.