Midterm 1 Applied Data Mining

Keith Hickman

November 14, 2017

Problem 1

```
## install.packages("data.table")
library(data.table)
library(ggplot2)
mydata <- read.csv("C:/Users/khickman/Desktop/Personal/IUMSDS/AppliedDataMining/Midterm/mydata.csv", se

summary(mydata)
```

```
##        V1                      V2            V3
##  Min.   :   1.0   ?           :   4   Min.   :-6.7749
##  1st Qu.: 500.8   -0.001405791:   1   1st Qu.:-2.2878
##  Median :1000.5   -0.002235545:   1   Median :-0.4438
##  Mean   :1000.5   -0.003699072:   1   Mean   :-0.9815
##  3rd Qu.:1500.2   -0.006583953:   1   3rd Qu.: 0.4354
##  Max.   :2000.0   -0.006972429:   1   Max.   : 3.1754
##                   (Other)     :1991   NA's   :8
##             V4              V5              X
##  ?             :   6   Min.   :-12.342   Min.   :1.00
##  -0.00303014 :   1   1st Qu.: -9.420   1st Qu.:1.75
##  -0.012157336:   1   Median : -8.628   Median :2.00
##  -0.017954776:   1   Mean   : -6.775   Mean   :1.75
##  -0.027248905:   1   3rd Qu.: -4.728   3rd Qu.:2.00
##  -0.031789989:   1   Max.   :  3.355   Max.   :2.00
##  (Other)     :1989
```

```
str(mydata)
```

```
## 'data.frame':    2000 obs. of  6 variables:
##  $ V1: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ V2: Factor w/ 1997 levels "-0.001405791",..: 1987 1330 1766 1850 1817 1768 1462 1870 1583 1809 ..
##  $ V3: num  -3.27 -3.94 -4.92 -2.79 -4.66 ...
##  $ V4: Factor w/ 1995 levels "-0.00303014",..: 745 746 942 889 662 742 809 1855 681 764 ...
##  $ V5: num  -9.21 -9.92 -8.66 -10.35 -10.58 ...
##  $ X : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
mydata[100:110,]
```

```
##      V1          V2        V3          V4         V5 X
## 100 100 2.106898626 -3.673282           ? -10.551039 1
## 101 101           ? -2.746142 9.093913921  -5.315355 1
## 102 102 3.326490963 -6.774908 9.702177633  -9.461351 1
## 103 103  2.33460081 -3.738696 10.36410186 -11.209379 1
## 104 104 4.167999798        NA 8.183001977  -9.487888 1
## 105 105 2.836025413 -3.413888 8.467948059  -8.823351 1
## 106 106 3.672512903 -3.648048 10.37116448  -8.959097 1
## 107 107 4.265924166 -3.897882  8.79822484  -9.610169 1
## 108 108 3.288826248 -2.518422 8.479622918  -8.470363 1
## 109 109 2.860199674 -3.332852 11.25037736  -8.391735 1
## 110 110 3.748196493 -2.918405 7.787542705  -8.632427 1
```

1.

There are 2000 observations of 6 variables.

2.

I noticed several missing values here, as well as some values with a `?` which is the same as NA. Additionally, I've got two of the variables that should be continuous listed as factors.

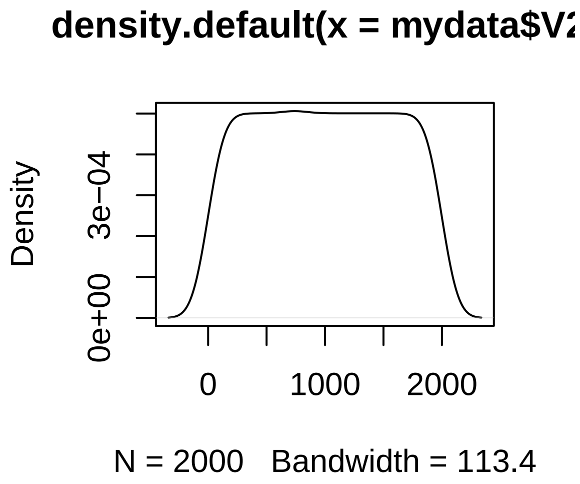
Starting with the factors:

```
mydata$V2 <- as.numeric(mydata$V2)
mydata$V4 <- as.numeric(mydata$V4)
str(mydata)
```

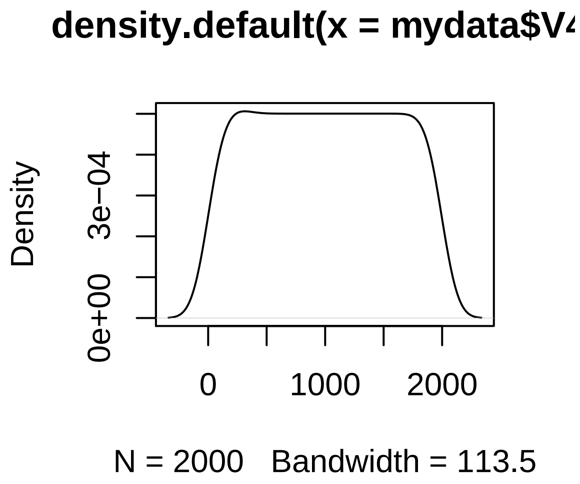
```
## 'data.frame':    2000 obs. of  6 variables:
##  $ V1: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ V2: num  1987 1330 1766 1850 1817 ...
##  $ V3: num  -3.27 -3.94 -4.92 -2.79 -4.66 ...
##  $ V4: num  745 746 942 889 662 ...
##  $ V5: num  -9.21 -9.92 -8.66 -10.35 -10.58 ...
##  $ X : int  1 1 1 1 1 1 1 1 1 1 ...
```

Now that the data columns are of the correct type, we can deal with missing or incorrect values:

```
plot(density(mydata$V2))
```



```
## plot(density(mydata$V3))
plot(density(mydata$V4))
```



I learned that only `V3` contains missing values, and that `V2` and `V4` are normal or mostly normally distributed, or at least symmetric.

```
## v3na <- is.na(mydata$V3)
v3na <- mydata[rowSums(is.na(mydata)) > 0,]
v3na
```

```
##      V1   V2 V3   V4         V5 X
## 50   50 1855 NA  908  -8.659472 1
## 70   70 1843 NA  653 -10.469044 1
## 104 104 1938 NA 1078  -9.487888 1
## 201 201 1912 NA 1923  -9.402905 1
## 301 301 1586 NA 1179  -9.106679 1
## 401 401 1631 NA 1658  -9.761055 1
## 800 800  619 NA  834  -9.125540 2
## 900 900 1588 NA 1639  -9.802796 2
```

Great - 8 rows of the V3 variable have NA values. I'll also check for question marks a bit later on. For now, let's impute the missing values. We'll use the mean because the shape of the variable indicates that most of the values are likely close to the central statistic.

```
mydata[50, "V3"] <- mean(mydata$V3, na.rm = TRUE)
mydata[70, "V3"] <- mean(mydata$V3, na.rm = TRUE)
mydata[104, "V3"] <- mean(mydata$V3, na.rm = TRUE)
mydata[201, "V3"] <- mean(mydata$V3, na.rm = TRUE)
mydata[301, "V3"] <- mean(mydata$V3, na.rm = TRUE)
mydata[401, "V3"] <- mean(mydata$V3, na.rm = TRUE)
mydata[800, "V3"] <- mean(mydata$V3, na.rm = TRUE)
mydata[900, "V3"] <- mean(mydata$V3, na.rm = TRUE)

summary(mydata$V3)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -6.7750 -2.2680 -0.4510 -0.9815  0.4299  3.1750
```

Great - looks like that did the trick. It imputed all values to the same number, however, so that's something that we may have to come back to later on.

Let's continue with problem 1.

3.

```
mean(mydata$V2)
```

```
## [1] 998.6115
```

```
median(mydata$V2)
```

```
## [1] 997.5
```

The values are very close together, especially considering the scale. This looks good for a normally distributed variable.

4.

```
var(mydata$V4)
```

```
## [1] 332438.5
```

```
sd(mydata$V4)
```

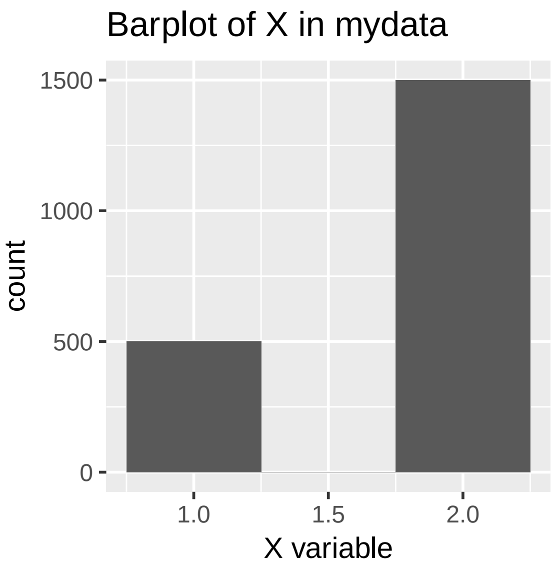
```
## [1] 576.5748
```

```
IQR(mydata$V4)
```

```
## [1] 999.5
```

Moving on to the barplot.

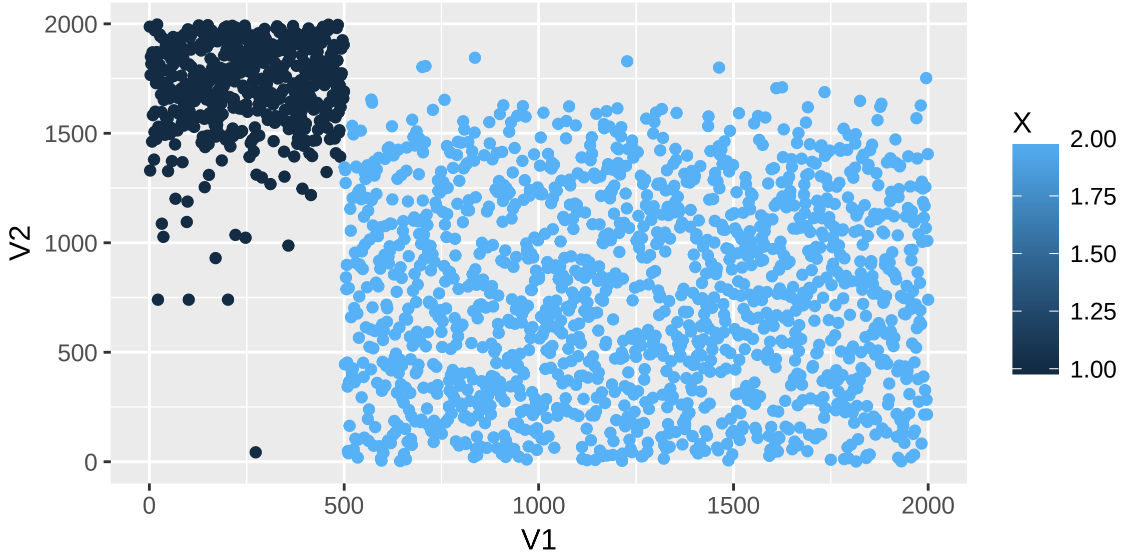
```
qplot(mydata$X, bins=3, xlab="X variable", main = "Barplot of X in mydata")
```



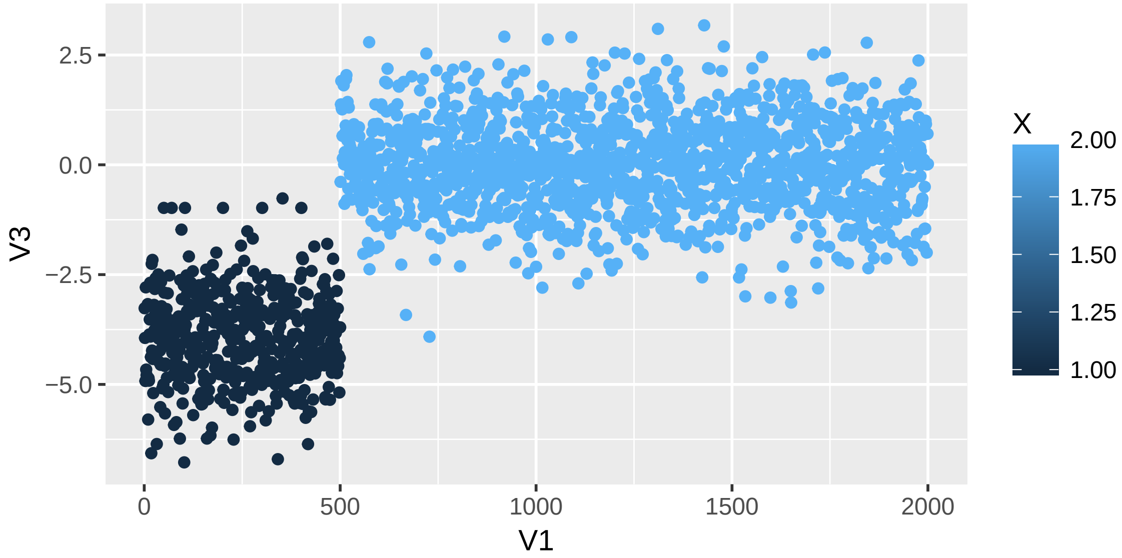
Looks like we have an uneven class distribution between class 1 and 2 in the X variable. This will likely be a factor in fitting models later on.

5.

```
qplot(V1, V2, data=mydata, colour=X)
```



```
qplot(V1, V3, data=mydata, colour=X)
```


2.5
0.0
−2.5
−5.0
V3
0
500
1000
1500
2000
V1
X
2.00
1.75
1.50
1.25
1.00