29339472_assignment_1_2020

August 9, 2020

In [1]:
```
library(MASS)
library(testthat)
library(tidyverse)
library(gclus)
library(GGally)


# Plot size deppening on your screen resolution to 4 x 3
options(repr.plot.width=6, repr.plot.height=4)
```

```
 Attaching packages  tidyverse 1.2.1
 ggplot2 3.1.0      purrr   0.2.5
 tibble  1.4.2      dplyr   0.7.5
 tidyr   0.8.1      stringr 1.3.1
 readr   1.1.1      forcats 0.3.0
 Conflicts  tidyverse_conflicts()
 dplyr::filter()  masks stats::filter()
 purrr::is_null() masks testthat::is_null()
 dplyr::lag()     masks stats::lag()
 dplyr::matches() masks testthat::matches()
 dplyr::select()  masks MASS::select()
Loading required package: cluster

Attaching package: GGally

The following object is masked from package:dplyr:

    nasa
```

1 Assignment 1

For this assignment, you have to analyse two different data sets. For each plot, add appropriate titles and labels, do not leave the default one. Your submission should run without errors on jupyterhub when you execute: `Kernel -> Restart & Run All`. For each data-set and task, assign

the resulting plot or data frame to the specified variable. e.g. if it says `p_3 <- ...` this means that the plot has to be assigned to variable `p_3`.

1.1 Dataset01 (3.5 points)

The dataset `birthwt` is in the package `MASS`, and will be available when you load the package. Have a quick look at the meaning of the variables in the data set `birthwt`.We will use `smoke` and `bwt`. 1. First create a new facotr variable called `smoke` and suplly labels for the levels of the variable smoke (Note that 0 means `nonsmoker`, 1 means `smoker`.) 2. Create a new data frame called `ds_a.df` with the factor `smoke` and the variable `bwt` (Hint the name should just be `bwt`) 2. Create a boxplot for the birthweights of the two groups: smokers and non-smokers. 3. Also provide a bar chart showing the number of smokers and non-smokers For both graphs, include a heading and appropriate axis labels.

From the Help file:
Risk Factors Associated with Low Infant Birth Weight Description

The birthwt data frame has 189 rows and 10 columns. The data were collected at Baystate Medical Center, Springfield, Mass during 1986.

In [2]:
```
## Dataset01 - Task 1 code here

# your code here

smoke <- factor(birthwt$smoke,levels = c(0,1),labels=c("nonsmoker","smoker"))
head(smoke)
```

1. nonsmoker 2. nonsmoker 3. smoker 4. smoker 5. smoker 6. nonsmoker
Levels: 1. 'nonsmoker' 2. 'smoker'

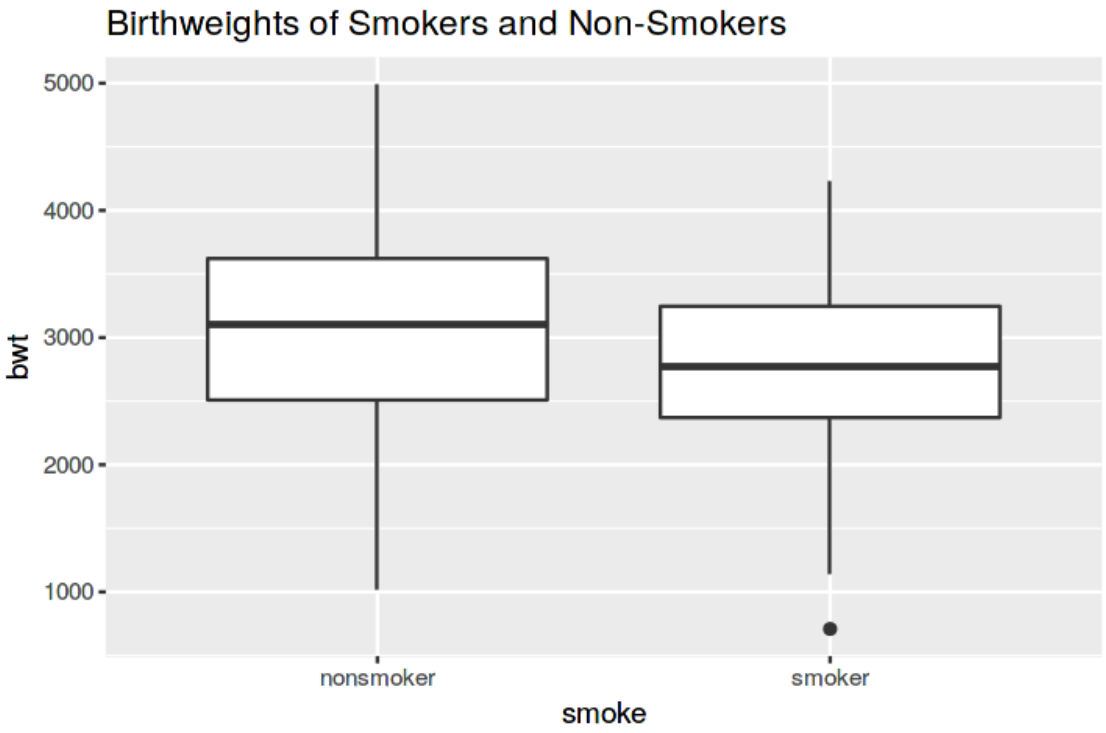
In [3]:
```
# this is the first public test to check that you assigned the correct variable, the res
test_that("Creating a factor variable", {
      expect_is(smoke,"factor")
})
```

In [4]:
```
## Dataset01 - Task 2 code here

# your code here

birthwt %>%
select(smoke,bwt) -> ds_a.df
ds_a.df
```

	smoke	bwt
85	0	2523
86	0	2551
87	1	2557
88	1	2594
89	1	2600
91	0	2622
92	0	2637
93	0	2637
94	1	2663
95	1	2665
96	0	2722
97	0	2733
98	0	2751
99	0	2750
100	1	2769
101	1	2769
102	0	2778
103	1	2782
104	0	2807
105	1	2821
106	0	2835
107	0	2835
108	0	2836
109	0	2863
111	0	2877
112	0	2877
113	1	2906
114	0	2920
115	1	2920
116	0	2920
44	1	2211
45	1	2225
46	0	2240
47	0	2240
49	0	2282
50	1	2296
51	1	2296
52	0	2301
54	0	2325
56	1	2353
57	0	2353
59	1	2367
60	1	2381
61	1	2381
62	0	2381
63	0	2410
65	1	2410
67	1	2410
68	1	2414
69	1	2424
71	0	2438

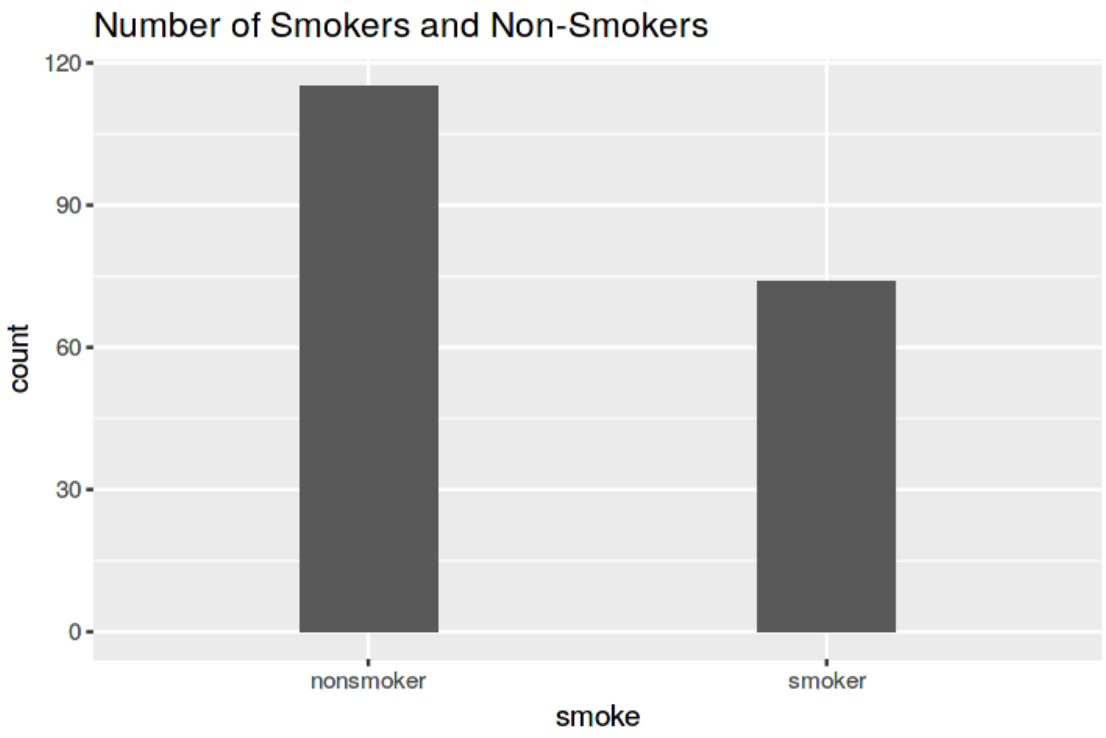
In [5]:
```
# This is the first public test to check that you assigned the correct variable, the res
test_that("Creating a data frame", {
      expect_is(ds_a.df,"data.frame")
})
```

In [6]:
```
## Dataset01 - Task 3 code here
# p_3 <- .....
# p_3

# your code here

p_3 <- ds_a.df %>%
mutate (smoke = factor(birthwt$smoke,levels = c(0,1),labels=c("nonsmoker","smoker"))) %>
ggplot(data=.,
           aes(x=smoke,
              y=bwt)) +
geom_boxplot()+ggtitle("Birthweights of Smokers and Non-Smokers")
p_3
```



In [7]:
```
# This is the first public test to check that you assigned the correct variable, the res
test_that("Boxplot", {
      expect_is(p_3,"ggplot")
})
```

In [8]:
```
## Dataset01 - Task 4 code here
# p_4 <- .....
# p_4
# your code here

p_4 <- ds_a.df %>%
mutate (smoke = factor(birthwt$smoke,levels = c(0,1),labels=c("nonsmoker","smoker"))) %>
    ggplot(data = .,
           aes(x = smoke)) +
    geom_bar(width=.3) + ggtitle("Number of Smokers and Non-Smokers")
p_4
```



In [9]:
```
# This is the first public test to check that you assigned the correct variable, the res
test_that("Bar plot", {
      expect_is(p_4,"ggplot")
})
```

1.2 Dataset02 (6.5 points)

Consider the data set bank in package gclus. In order to find information about this data set, look up the helpfile `?bank` You will see that it has a variable Status, and six dimension variables. The dataset must be loaded: with the command `data(bank)` 1. Load the data and make the column `Status` into a factor with `0 = Genuine` and `1 = Counterfeit`. (Note: modify the data frame) 2. Create a boxplot for each dimension (`Length`, `Left`,`Right`,`Bottom`, `Top` and `Diagonal`). Based

on the boxplots, choose two variables that you think likely to give the clearest differentiation between forged and genuine notes and put the graphs object named `p_var1` and `p_var2` in the designated cell. 3. Create a scatter plot of the choosen dimension and colour them. Use the function `geom_segment` to add a seperating line between the coloured dots. Note: This is a new function, the goal of this excercise is that you make yourself familiar with how to use the help function. 4. Using a grammar of graphics command, create a scatterplot matrix in which the points representing the forgeries and the genuine notes have different colours. Also had a title to your plot.

5. Using a grammar of graphics command, create a scatterplot matrix for the combined sample of 200 notes, including the overall distribution for each variable provided along the diagonal, and correlations in the upper panel. Also had a title to your plot.

6. Use ggcorrplot to create an appropriate correlation plot for the combined sample. Also had a title to your plot. Don't forgett to include the library installation and loading comands.

From the Help file:
Swiss bank notes data Description

Data from "Multivariate Statistics A practical approach", by Bernhard Flury and Hans Riedwyl, Chapman and Hall, 1988, Tables 1.1 and 1.2 pp. 5-8. Six measurements made on 100 genuine Swiss banknotes and 100 counterfeit ones.

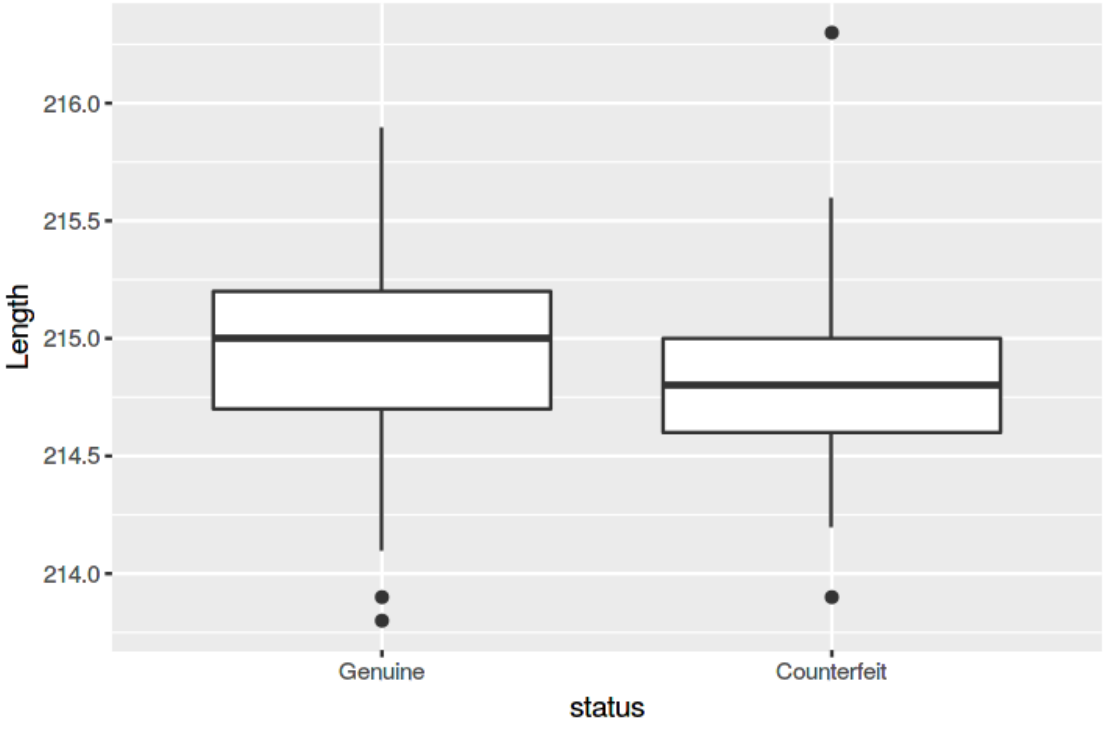
In [10]:
```
## Dataset02 - Task 1 code here
# your code here
data(bank)
Status <- factor(bank$Status,levels = c(0,1),labels=c("Genuine","Counterfeit"))
head(Status)
```

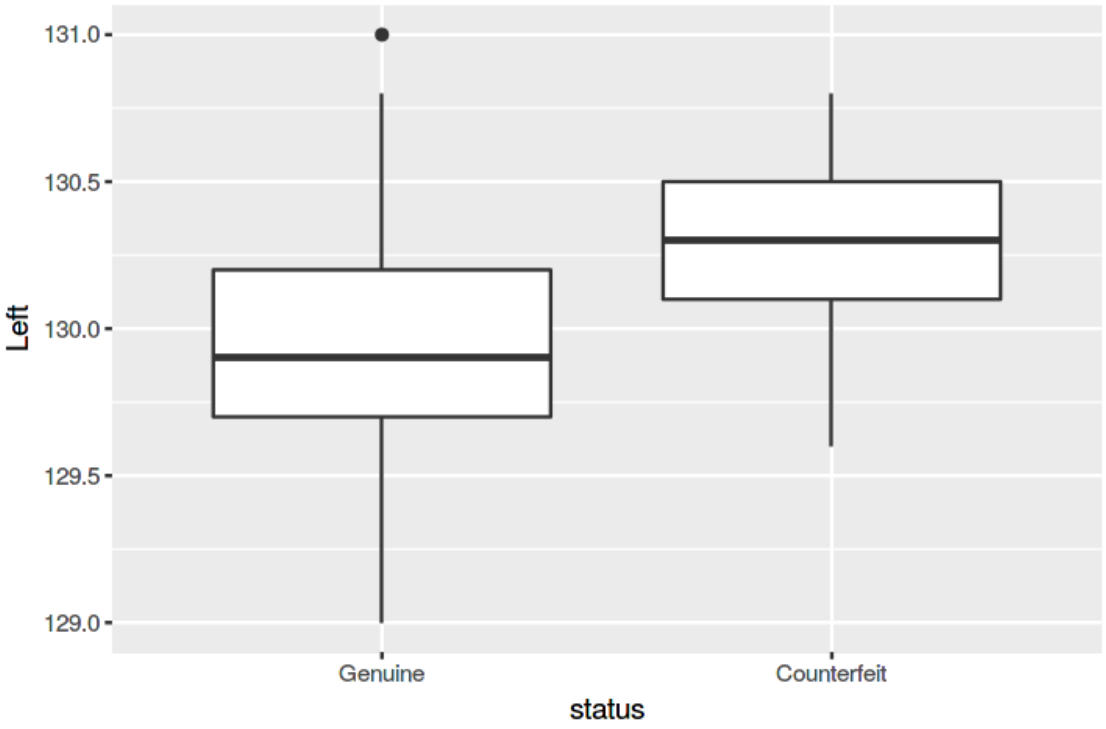
1. Genuine 2. Genuine 3. Genuine 4. Genuine 5. Genuine 6. Genuine
Levels: 1. 'Genuine' 2. 'Counterfeit'

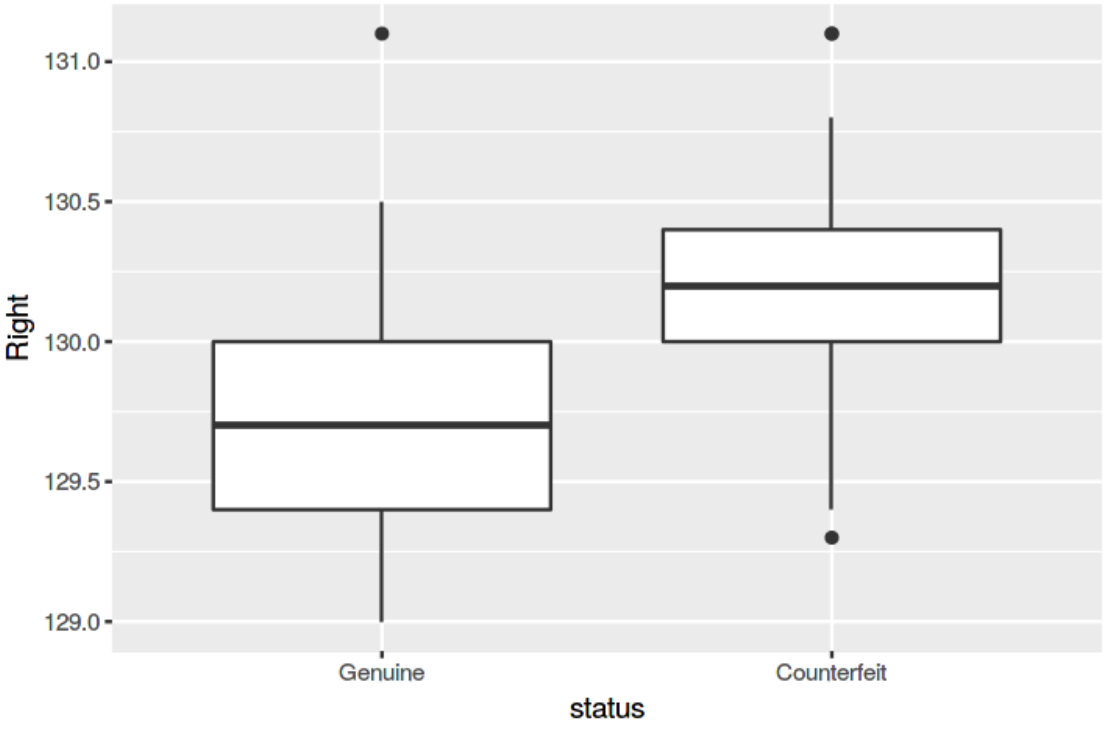
In [11]:
```
# this is the first public test to check that you assigned the correct variable, the re
test_that("Creating a factor variable within a existing data frame", {
      expect_is(bank,"data.frame")
})
```

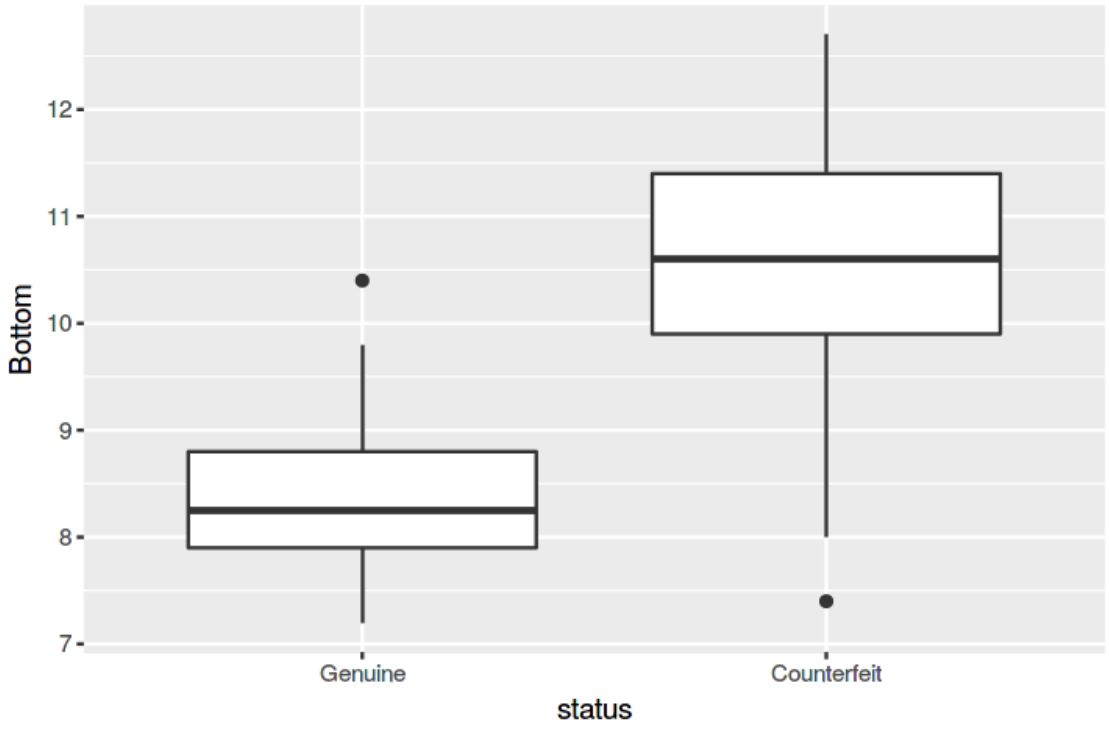
In [12]:
```
## Dataset02 - Task 2 goes here

plength <- bank %>%
mutate (status = factor(bank$Status,levels = c(0,1),labels=c("Genuine","Counterfeit")))
ggplot(data=.,aes(x=status,y=Length)) + geom_boxplot()
plength

pleft <- bank %>%
mutate (status = factor(bank$Status,levels = c(0,1),labels=c("Genuine","Counterfeit")))
ggplot(data=.,aes(x=status,y=Left)) + geom_boxplot()
pleft

pright <- bank %>%
```

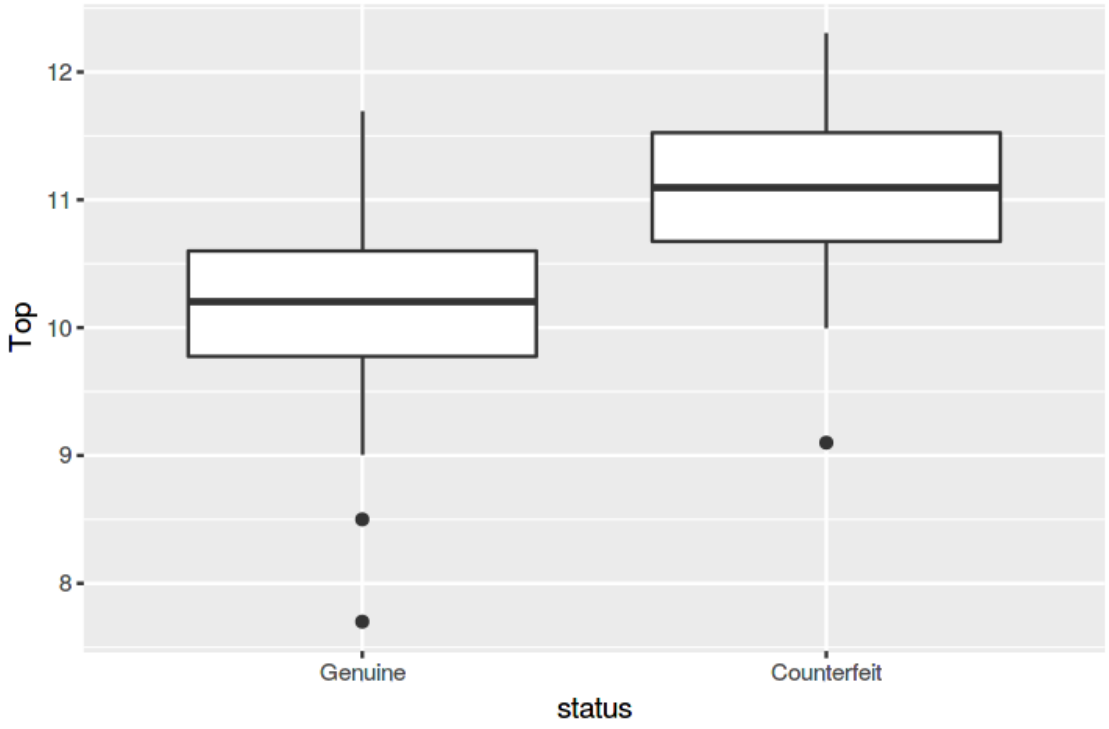
```
mutate (status = factor(bank$Status,levels = c(0,1),labels=c("Genuine","Counterfeit")))
ggplot(data=.,aes(x=status,y=Right)) + geom_boxplot()
pright

pbottom <- bank %>%
mutate (status = factor(bank$Status,levels = c(0,1),labels=c("Genuine","Counterfeit")))
ggplot(data=.,aes(x=status,y=Bottom)) + geom_boxplot()
pbottom

ptop <- bank %>%
mutate (status = factor(bank$Status,levels = c(0,1),labels=c("Genuine","Counterfeit")))
ggplot(data=.,aes(x=status,y=Top)) + geom_boxplot()
ptop

pdiagonal <- bank %>%
mutate (status = factor(bank$Status,levels = c(0,1),labels=c("Genuine","Counterfeit")))
ggplot(data=.,aes(x=status,y=Diagonal)) + geom_boxplot()
pdiagonal
```

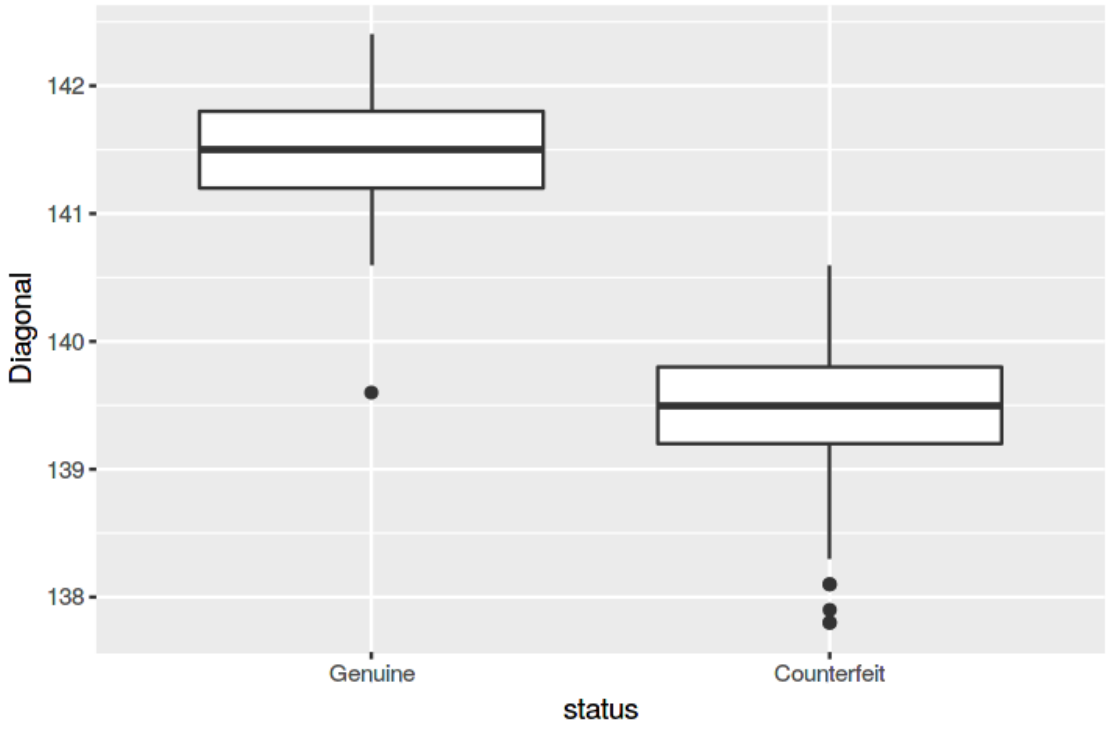



131.0
130.5
130.0
129.5
129.0
Left
Genuine
Counterfeit
status

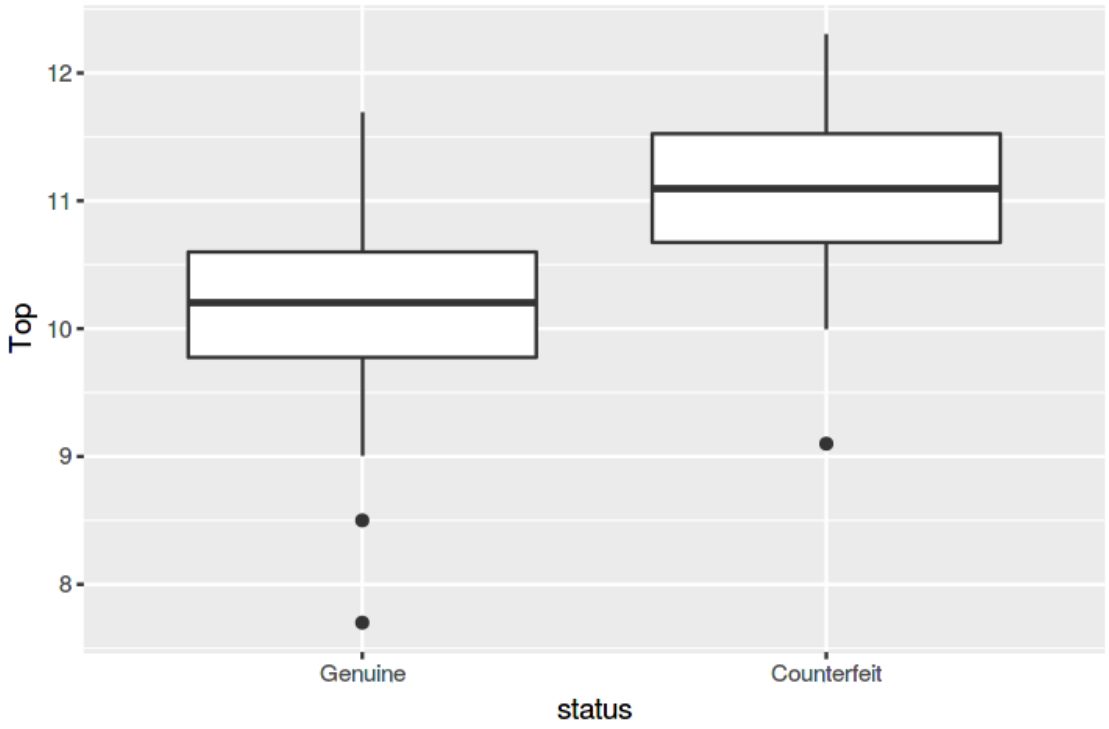

131.0
130.5
130.0
129.5
129.0
Right
Genuine
Counterfeit
status

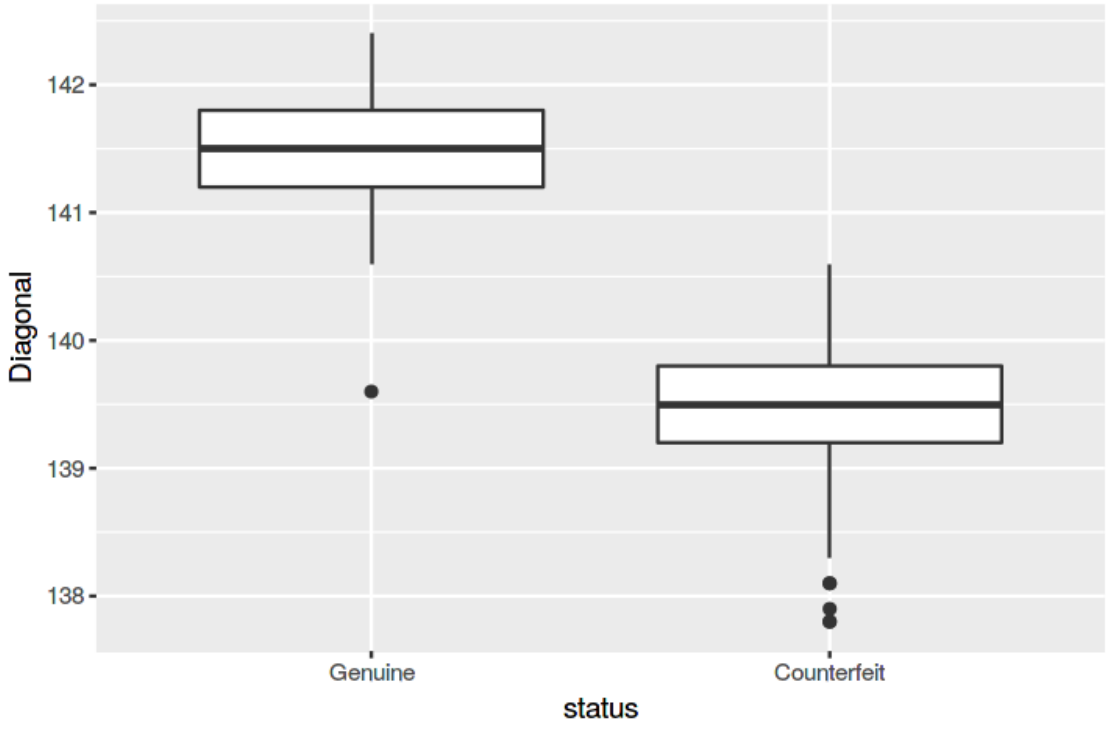

Bottom
12
11
10
9
8
7
Genuine
Counterfeit
status


Top
12
11
10
9
8
Genuine
Counterfeit
status



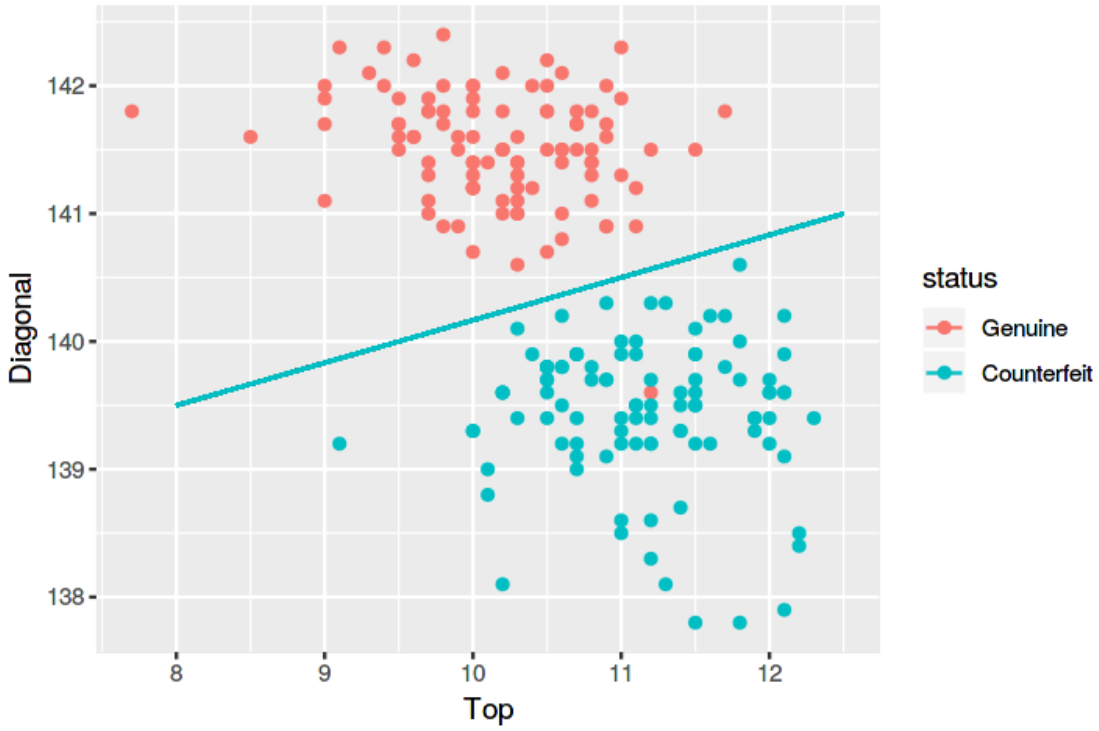
In [13]:
```
## Dataset02 - Task 2 goes here
# your code here
## I think Top and Diagonal dimensions likely to give the clearest differentiation betu
p_var1 <- bank %>%
mutate (status = factor(bank$Status,levels = c(0,1),labels=c("Genuine","Counterfeit")))
ggplot(data=.,aes(x=status, y=Top)) + geom_boxplot()
p_var1

p_var2 <- bank %>%
mutate (status = factor(bank$Status,levels = c(0,1),labels=c("Genuine","Counterfeit")))
ggplot(data=.,aes(x=status,y=Diagonal)) + geom_boxplot()
p_var2
```


12
11
10
9
8
Top
Genuine
Counterfeit
status


142
141
140
139
138
Diagonal
Genuine
Counterfeit
status

In [14]:
```
# This is the first public test to check that you assigned the correct variable, the re
test_that("Boxplot", {
      expect_is(p_var1,"ggplot")
      expect_is(p_var2,"ggplot")

})
```

In [15]:
```
## Dataset02 - Task 3 code here
# your code here
p_3 <- bank %>%
mutate (status = factor(bank$Status,levels = c(0,1),labels=c("Genuine","Counterfeit")))
ggplot(data=.,
           aes(x=Top,
              y=Diagonal,
              colour=status)) +
geom_point() + geom_segment(aes(x=8,y=139.5,xend=12.5,yend=141))
p_3
```

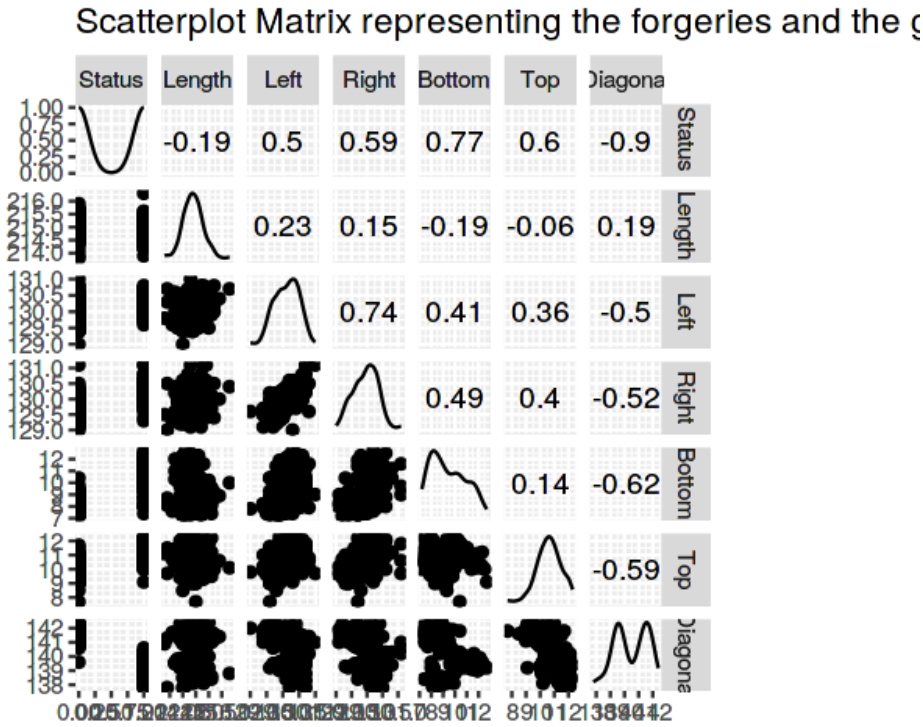


In [16]:
```
## Dataset02 - Task 4 code here
# your code here
#options(repr.plot.width=6, repr.plot.height=4)
bank %>%
mutate (status = factor(bank$Status,levels = c(0,1),labels=c("Genuine","Counterfeit")))
```

```
bankupd
mat <- ggscatmat(data=bankupd)+ ggtitle("Scatterplot Matrix representing the forgeries
mat
```

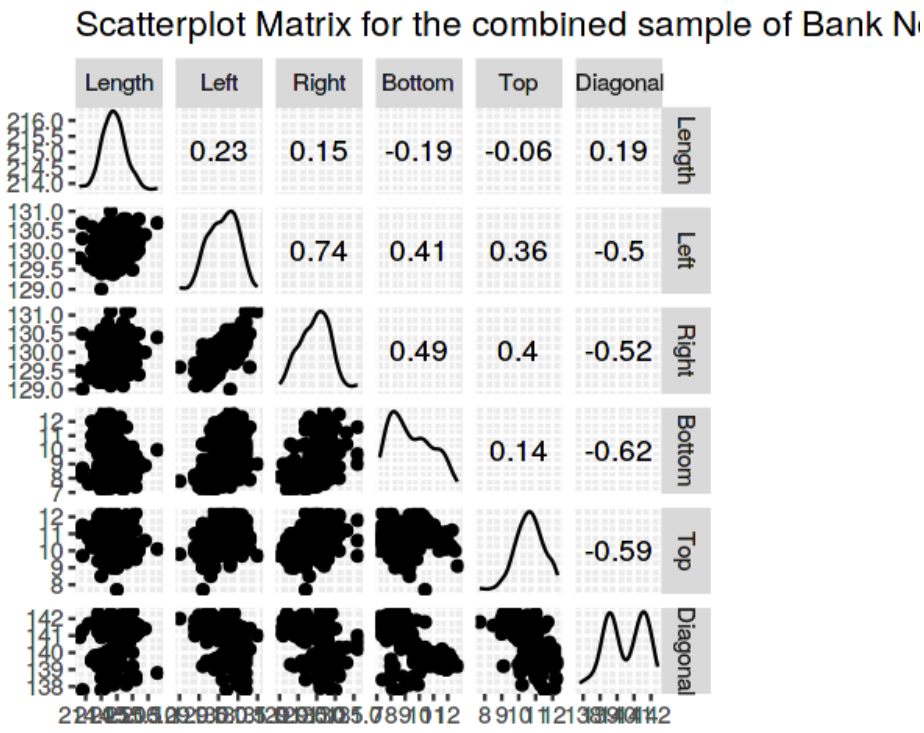
Status	Length	Left	Right	Bottom	Top	Diagonal	status
0	214.8	131.0	131.1	9.0	9.7	141.0	Genuine
0	214.6	129.7	129.7	8.1	9.5	141.7	Genuine
0	214.8	129.7	129.7	8.7	9.6	142.2	Genuine
0	214.8	129.7	129.6	7.5	10.4	142.0	Genuine
0	215.0	129.6	129.7	10.4	7.7	141.8	Genuine
0	215.7	130.8	130.5	9.0	10.1	141.4	Genuine
0	215.5	129.5	129.7	7.9	9.6	141.6	Genuine
0	214.5	129.6	129.2	7.2	10.7	141.7	Genuine
0	214.9	129.4	129.7	8.2	11.0	141.9	Genuine
0	215.2	130.4	130.3	9.2	10.0	140.7	Genuine
0	215.3	130.4	130.3	7.9	11.7	141.8	Genuine
0	215.1	129.5	129.6	7.7	10.5	142.2	Genuine
0	215.2	130.8	129.6	7.9	10.8	141.4	Genuine
0	214.7	129.7	129.7	7.7	10.9	141.7	Genuine
0	215.1	129.9	129.7	7.7	10.8	141.8	Genuine
0	214.5	129.8	129.8	9.3	8.5	141.6	Genuine
0	214.6	129.9	130.1	8.2	9.8	141.7	Genuine
0	215.0	129.9	129.7	9.0	9.0	141.9	Genuine
0	215.2	129.6	129.6	7.4	11.5	141.5	Genuine
0	214.7	130.2	129.9	8.6	10.0	141.9	Genuine
0	215.0	129.9	129.3	8.4	10.0	141.4	Genuine
0	215.6	130.5	130.0	8.1	10.3	141.6	Genuine
0	215.3	130.6	130.0	8.4	10.8	141.5	Genuine
0	215.7	130.2	130.0	8.7	10.0	141.6	Genuine
0	215.1	129.7	129.9	7.4	10.8	141.1	Genuine
0	215.3	130.4	130.4	8.0	11.0	142.3	Genuine
0	215.5	130.2	130.1	8.9	9.8	142.4	Genuine
0	215.1	130.3	130.3	9.8	9.5	141.9	Genuine
0	215.1	130.0	130.0	7.4	10.5	141.8	Genuine
0	214.8	129.7	129.3	8.3	9.0	142.0	Genuine
1	213.9	130.7	130.5	8.7	11.5	137.8	Counterfeit
1	214.2	130.6	130.4	12.0	10.2	139.6	Counterfeit
1	214.8	130.5	130.3	11.8	10.5	139.4	Counterfeit
1	214.8	129.6	130.0	10.4	11.6	139.2	Counterfeit
1	214.8	130.1	130.0	11.4	10.5	139.6	Counterfeit
1	214.9	130.4	130.2	11.9	10.7	139.0	Counterfeit
1	214.3	130.1	130.1	11.6	10.5	139.7	Counterfeit
1	214.5	130.4	130.0	9.9	12.0	139.6	Counterfeit
1	214.8	130.5	130.3	10.2	12.1	139.1	Counterfeit
1	214.5	130.2	130.4	8.2	11.8	137.8	Counterfeit
1	215.0	130.4	130.1	11.4	10.7	139.1	Counterfeit
1	214.8	130.6	130.6	8.0	11.4	138.7	Counterfeit
1	215.0	130.5	130.1	11.0	11.4	139.3	Counterfeit
1	214.6	130.5	130.4	10.1	11.4	139.3	Counterfeit
1	214.7	130.2	130.1	10.7	11.1	139.5	Counterfeit
1	214.7	130.4	130.0	11.5	10.7	139.4	Counterfeit
1	214.5	130.4	130.0	8.0	12.2	138.5	Counterfeit
1	214.8	130.0	129.7	11.4	10.6	139.2	Counterfeit
1	214.8	129.9	130.2	9.6	11.9	139.4	Counterfeit
1	214.6	130.3	130.2	12.7	9.1	139.2	Counterfeit
1	215.1	130.2	129.8	10.2	12.0	139.4	Counterfeit

```
Warning message in ggscatmat(data = bankupd):
Factor variables are omitted in plot
```



In [17]:
```
# This is the first public test to check that you assigned the correct variable, the re
test_that("Plot", {
      expect_is(mat,"ggplot")

})
```

In [18]:
```
## Dataset02 - Task 5 code here
# your code here
data(bank)
mat_all <- ggscatmat(data=bank,columns=2:7)+ggtitle("Scatterplot Matrix for the combine
mat_all
```




```
In [19]: # This is the first public test to check that you assigned the correct variable, the re
         test_that("Plot", {
               expect_is(mat_all,"ggplot")

         })
```

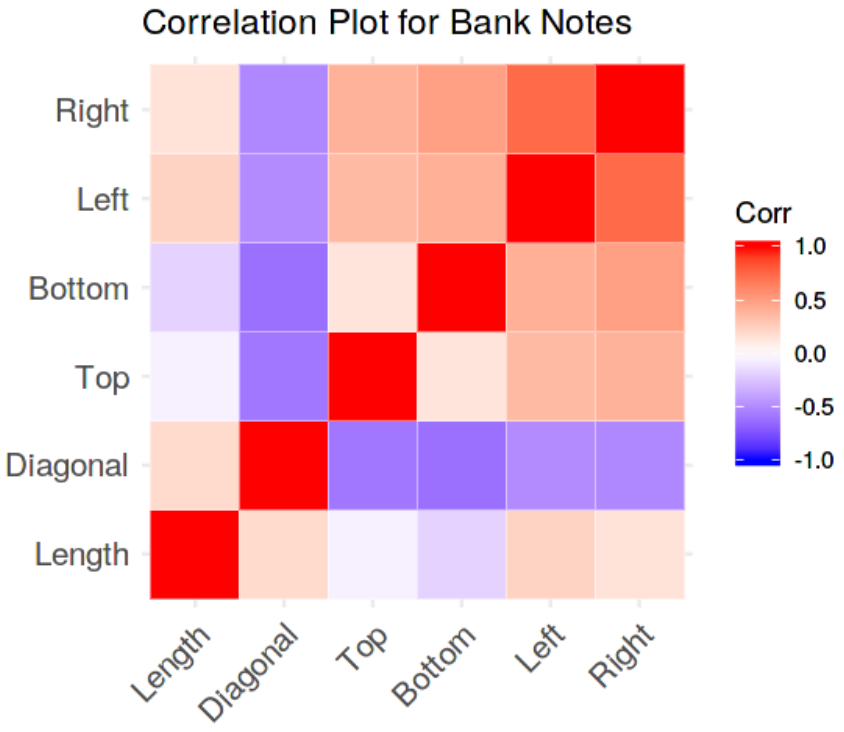
```
In [20]: install.packages('ggcorrplot',lib='.', verbose=TRUE)
```

```
system (cmd0): /usr/lib/R/bin/R CMD INSTALL
foundpkgs: ggcorrplot, /tmp/RtmpkX2oKH/downloaded_packages/ggcorrplot_0.1.3.tar.gz
files: /tmp/RtmpkX2oKH/downloaded_packages/ggcorrplot_0.1.3.tar.gz
1): succeeded '/usr/lib/R/bin/R CMD INSTALL -l '/srv/home/adey0001' /tmp/RtmpkX2oKH/downloaded_p
```

```
In [21]: library(ggcorrplot,lib.loc='.')
```

```
In [22]: ## Dataset02 - Task 6 code here
         corr <- cor(bank[,2:7])
         corr
         p_cor <- ggcorrplot(corr,hc.order = TRUE,outline.col = "white")+ggtitle("Correlation Pl
         p_cor
```

	Length	Left	Right	Bottom	Top	Diagonal
Length	1.00000000	0.2312926	0.1517628	-0.1898009	-0.06132141	0.1943015
Left	0.23129257	1.0000000	0.7432628	0.4137810	0.36234960	-0.5032290
Right	0.15176280	0.7432628	1.0000000	0.4867577	0.40067021	-0.5164755
Bottom	-0.18980092	0.4137810	0.4867577	1.0000000	0.14185134	-0.6229827
Top	-0.06132141	0.3623496	0.4006702	0.1418513	1.00000000	-0.5940446
Diagonal	0.19430146	-0.5032290	-0.5164755	-0.6229827	-0.59404464	1.0000000



In [23]:
```
# This is the first public test to check that you assigned the correct variable, the re
test_that("Scater Plot", {
      expect_is(p_cor,"ggplot")
})
```