Exercise 2

Let's use the dataset we discussed in Assignment 1 (TeleCustomers.xlsx, copyright © IBM Academic Initiative Program). The dataset contains the following fields describing the customers.

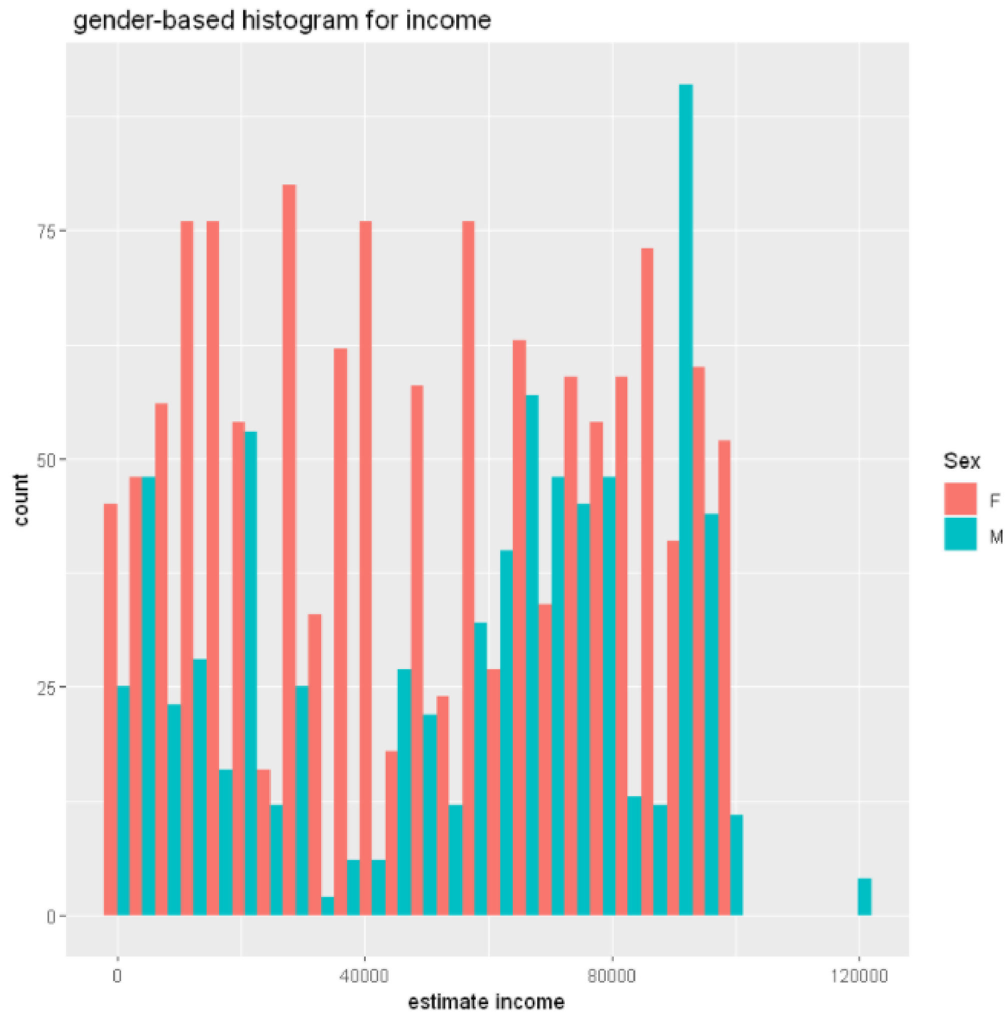
ID	Customer reference number
Sex	Gender
Status	Marital status
Children	Number of Children
Est_Income	Estimated income
Car_Owner	Car owner
Usage	Time spent on calls in total per month
Age	Age
RatePlan	Chosen rate plan (1, 2, …)
LongDistance	Time spent on long distance calls per month
International	Time spent on international calls per month
Local	Time spent on local calls per month
Dropped	Number of dropped calls
Payment	Payment method of the monthly telephone bill
LocalBillType	Tariff for locally based calls
LongDistanceBillType	Tariff for long distance calls
CHURNED	Current vs. Cancelled

1. (10 points) Plot gender-based histograms to compare the "Estimated income" and "Usage" respectively.

1. Plotting gender-based histograms to compare the "Estimated income" and "Usage" respectively.

Gender-based histograms to compare the Estimated income

```
In [42]:
1 ggplot(data=data, aes(Est_Income, fill=Sex)) + geom_histogram(bins = 30, position = position_dodge())
2 +xlab("estimate income")+ ggtitle(" gender-based histogram for income")
3
```



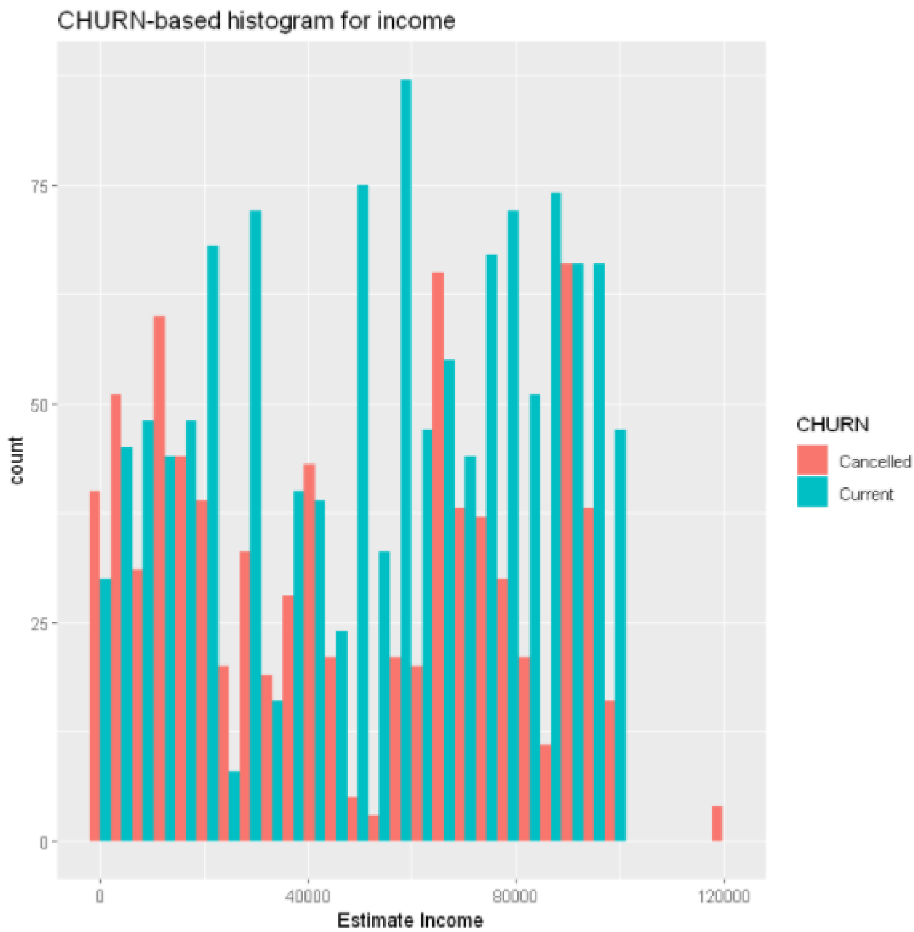
According to the graph we can see there are a greater number of females with estimated salary between 0 to 85k, whereas more males with salary range of 90k+

2. (10 points) Plot CHURNED-based histograms to compare the "Estimated income" and "Usage" respectively.

2. Plotting CHURNED-based histograms to compare the "Estimated income" and "Usage" respectively.

CHURN-based histogram for income

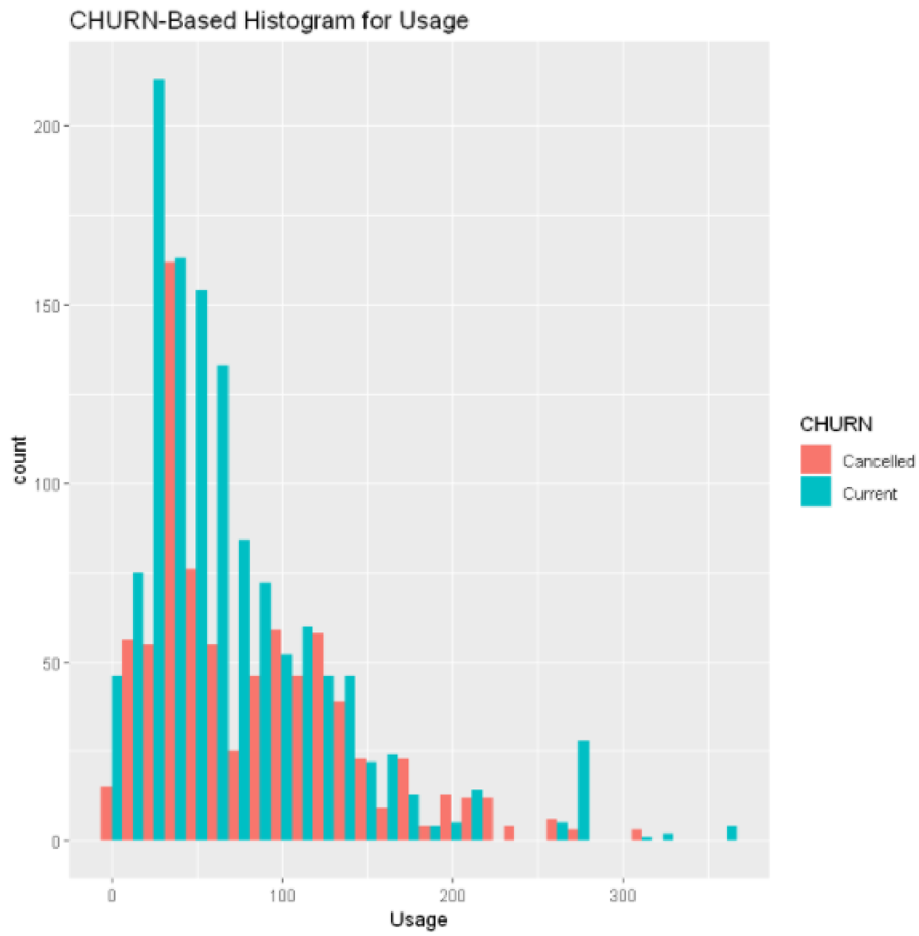
```
ggplot(data=data, aes(Est_Income, fill=CHURN)) + geom_histogram(bins =30,position = position_dodge())+xlab("Estimate Income"
```



In general we can conculed for 20k to 60k estimate income the number of cancelled users/ churned users are quite low

CHURN-Based Histogram for Usage

```
]:  1  ggplot(data=data, aes(Usage, fill=CHURN)) + geom_histogram(bins = 30,position = position_dodge())+xlab("Usage")+ ggtitle("CH
    2
```

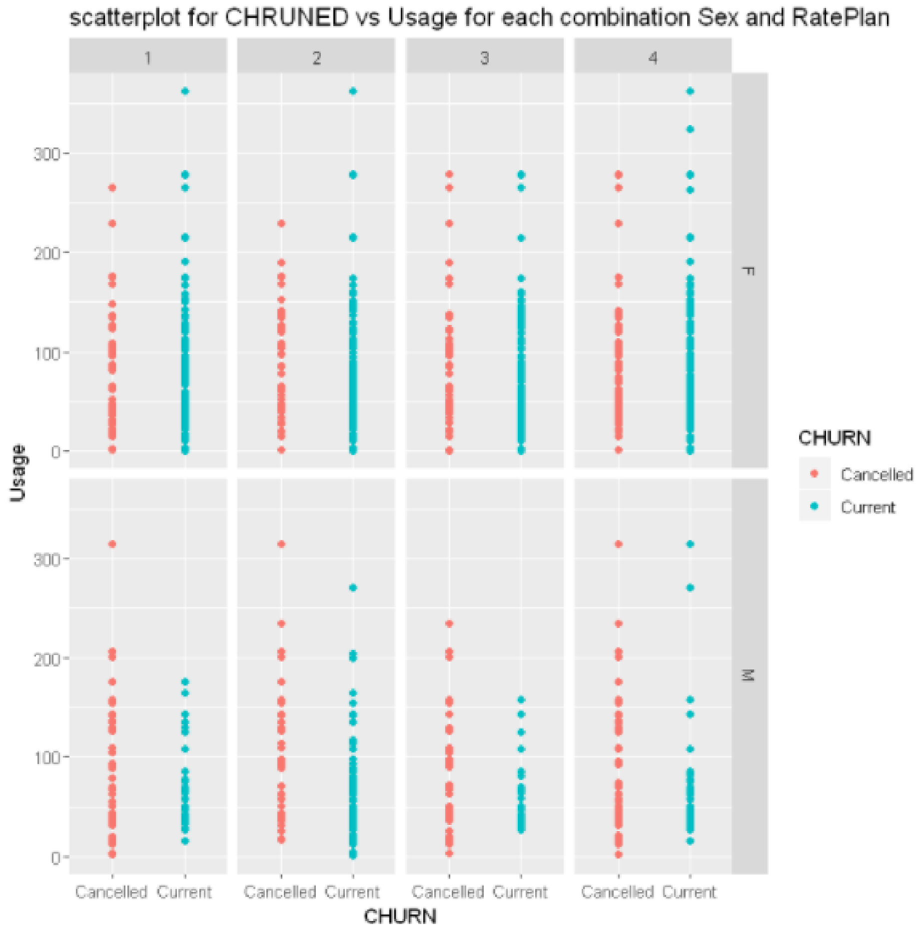


From this we CANNOT conclude that more or lesser the usage more the churn ratio

3. (10 points) Use scatterplots in matrix form to show CHRUNED vs Usage for each combination Sex and RatePlan (please refer to Figure 3.2.2).

3. scatterplots in matrix form to show CHRUNED vs Usage for each combination Sex and RatePlan

In [46]:
```
ggplot(data, aes(x=CHURN,y=Usage, group=CHURN)) + geom_point(aes(color=CHURN)) + ylab("Usage") + xlab("CHURN") + facet_grid(
```

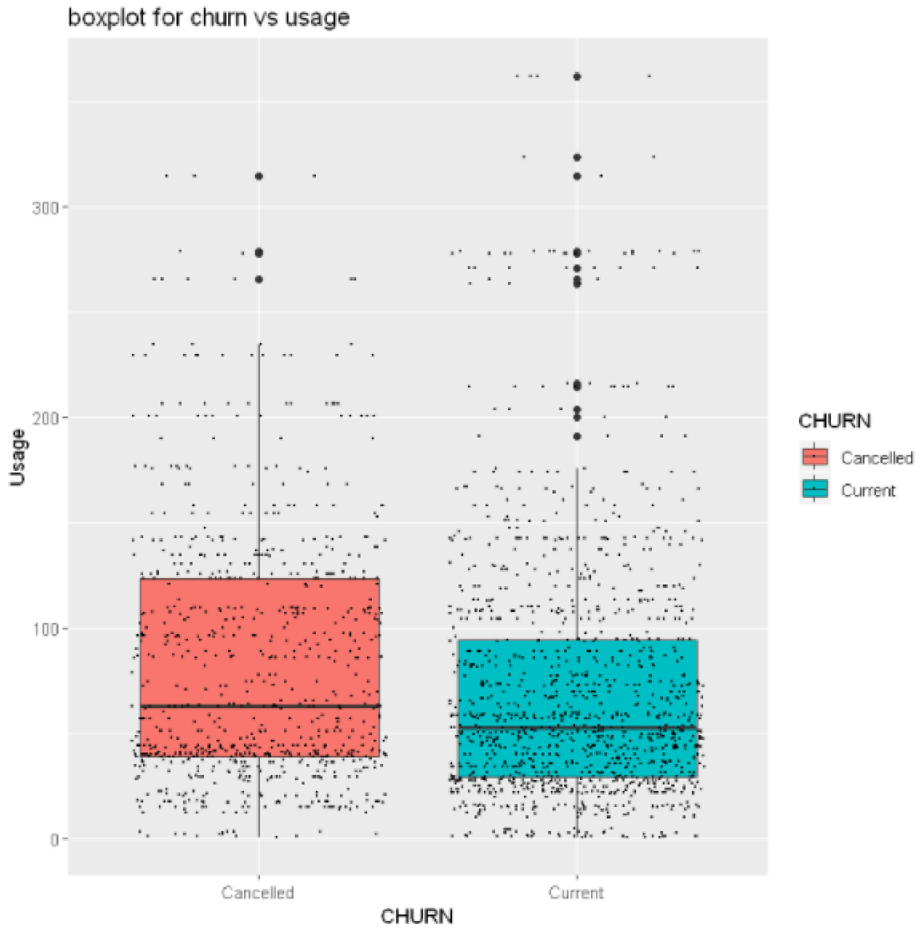


rateplan 1 and 3 have more scattered usage in females than in males for people who are still currently using the services

4. (10 points) Use boxplot to show CHRUNED vs Usage (please refer to Figure 3.2.3).

4. boxplot to show CHRUNED vs Usage

```
1 ggplot(data, aes(x = CHURN, y = Usage, fill=CHURN)) + geom_boxplot(aes()) + ggtitle("boxplot for churn vs usage")+geom_jitt
```



From usage alone it is tough to infer if the customer will churn or not as their is not a huge difference between the

Please use second Excel file (Hackathon) to create a data frame. Loan_Status indicates the approval of each loan application: Y for approved and N for declined.

Credits: https://www.analyticsvidhya.com/blog/2017/02/introduction-to-ensembling-along-with-implementation-in-r/

Ref: Caret => https://topepo.github.io/caret/index.html

```
#Loading the required libraries
library('caret')
#Seeting the random seed
set.seed(100)
#Loading the hackathon dataset
data_loanapp<-read.csv(url('https://datahack-prod.s3.ap-south-1.amazonaws.com/train_file/train_u6lujuX_CVtuZ9i.csv')) #Load directly from the URL
OR
2. Download the Hackathon file in your Directory and load it to a data frame

#Let's check the data structure of the loaded dataset
str(data_loanapp)
```

5. (12) Explore the data frame, identify and report the missing data. How will you deal with the missing data?

5. Exploring the Hackathon dataset

```
]:  1  head(d2)
```

...1	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_Histor
1	LP001002	Male	No	0	Graduate	No	5849	0	NA	360	
2	LP001003	Male	Yes	1	Graduate	No	4583	1508	128	360	
3	LP001005	Male	Yes	0	Graduate	Yes	3000	0	66	360	
4	LP001006	Male	Yes	0	Not Graduate	No	2583	2358	120	360	
5	LP001008	Male	No	0	Graduate	No	6000	0	141	360	
6	LP001011	Male	Yes	2	Graduate	Yes	5417	4196	267	360	

Dimensions of dataset

```
]:  1  dim(d2)
```

614 14

Lets glimplse over all the different type of values in each column

```
]:  1  library(dplyr)
    2  glimpse(d2)
```

```
Rows: 614
Columns: 14
$ ...1              <chr> "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "~
$ Loan_ID           <chr> "LP001002", "LP001003", "LP001005", "LP001006", "LP0~
$ Gender            <chr> "Male", "Male", "Male", "Male", "Male", "Male", "Mal~
$ Married           <chr> "No", "Yes", "Yes", "Yes", "No", "Yes", "Yes", "Yes"~
$ Dependents        <chr> "0", "1", "0", "0", "0", "2", "0", "3+", "2", "1", "~
$ Education         <chr> "Graduate", "Graduate", "Graduate", "Not Graduate", ~
$ Self_Employed     <chr> "No", "No", "Yes", "No", "No", "Yes", "No", "No", "N~
$ ApplicantIncome   <dbl> 5849, 4583, 3000, 2583, 6000, 5417, 2333, 3036, 4006~
$ CoapplicantIncome <dbl> 0, 1508, 0, 2358, 0, 4196, 1516, 2504, 1526, 10968, ~
$ LoanAmount        <dbl> NA, 128, 66, 120, 141, 267, 95, 158, 168, 349, 70, 1~
$ Loan_Amount_Term  <dbl> 360, 360, 360, 360, 360, 360, 360, 360, 360, 360, 36~
$ Credit_History    <dbl> 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, NA, ~
$ Property_Area     <chr> "Urban", "Rural", "Urban", "Urban", "Urban", "Urban"~
$ Loan_Status       <chr> "Y", "N", "Y", "Y", "Y", "Y", "Y", "N", "Y", "N", "Y~
```

Summary indicates the lenght class min max median and quartilies of each column

```
]: 1 summary(d2)
```

```
     ...1              Loan_ID             Gender              Married
 Length:614         Length:614         Length:614         Length:614
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character



  Dependents         Education         Self_Employed      ApplicantIncome
 Length:614         Length:614         Length:614         Min.   :  150
 Class :character   Class :character   Class :character   1st Qu.: 2878
 Mode  :character   Mode  :character   Mode  :character   Median : 3812
                                                          Mean   : 5403
                                                          3rd Qu.: 5795
                                                          Max.   :81000

 CoapplicantIncome   LoanAmount    Loan_Amount_Term Credit_History
 Min.   :    0     Min.   :  9.0   Min.   : 12      Min.   :0.0000
 1st Qu.:    0     1st Qu.:100.0   1st Qu.:360      1st Qu.:1.0000
 Median : 1188     Median :128.0   Median :360      Median :1.0000
 Mean   : 1621     Mean   :146.4   Mean   :342      Mean   :0.8422
 3rd Qu.: 2297     3rd Qu.:168.0   3rd Qu.:360      3rd Qu.:1.0000
 Max.   :41667     Max.   :700.0   Max.   :480      Max.   :1.0000
                   NA's   :22      NA's   :14       NA's   :50
 Property_Area       Loan_Status
 Length:614         Length:614
 Class :character   Class :character
 Mode  :character   Mode  :character
```

Identifying the missing data

```
In [52]: 1 sum(is.na(d2))
```

149

```
In [53]: 1 rowSums(is.na(d2))
```

```
1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 2 0 0 0 1 2 0 0 0 0 1 2 0 0 0 0 1 1 0 0 0 0 0 1 0 1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 1 0 0 1 0 0 0
0 0 0 0 0 3 0 0 0 0 0 0 2 1 2 0 0 1 0 0 0 1 1 1 1 0 0 1 0 0 1 0 0 0 0 1 1 1 0 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0
0 1 0 2 0 0 1 1 0 0 0 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 1
0 0 1 1 0 0 0 1 1 0 0 0 0 0 0 0 0 1 1 1 2 1 0 1 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0
0 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 3 0 1 0 0 0 0 0 0 1 0 0 1 0 1
0 1 0 0 0 0 0 0 0 0 2 0 0 1 0 0 0 1 1 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0
0 0 1 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0
0 0 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 1 0
0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 2 1 0 0 0 1 0 0 0 0 0 0 0 0
```

```
In [54]: 1 colSums(is.na(d2))
```

...1	0
Loan_ID	0
Gender	13
Married	3
Dependents	15
Education	0
Self_Employed	32
ApplicantIncome	0
CoapplicantIncome	0
LoanAmount	22
Loan_Amount_Term	14
Credit_History	50
Property_Area	0
Loan_Status	0

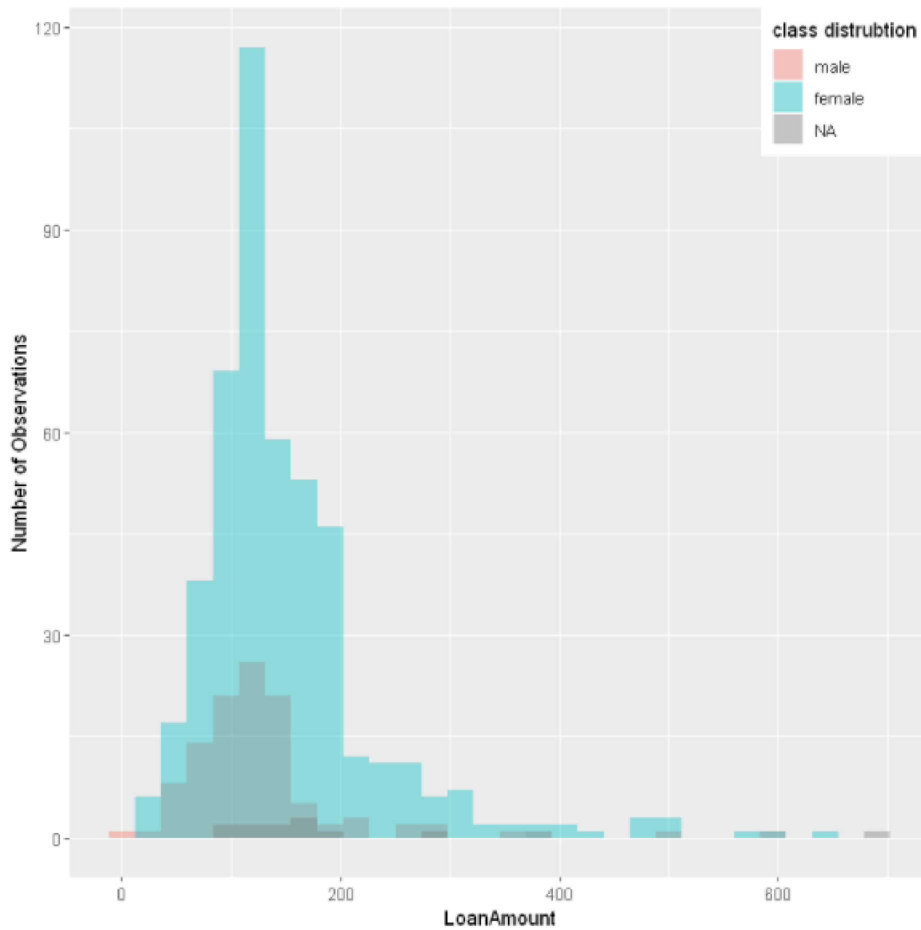
These are the columns that contain missing values in the corresponding numbers

Checking class distribution(Gender) wrt to loanamount

In [55]:
```
p <- ggplot( d2, aes( x=LoanAmount, fill=Gender))+
                            geom_histogram(bins = 30,alpha=0.4,position="identity")+
                            xlab("LoanAmount")+
                            scale_y_continuous("Number of Observations")

p + guides(fill=guide_legend(title="class distrubtion")) + scale_fill_discrete(labels=c("male","female")) +theme(legend.posi
```

```
Warning message:
"Removed 22 rows containing non-finite values (stat_bin)."
```



Gender class is highly skewed hence we cannot impute the missing gender categorical values with Mode

We can deal with the missing values in multiple ways:

1. Deleting the rows with missing values if the total number of rows are far higher than the missing value rows
2. Deleting the entire column if the number of missing values in a column is very high (close to number of total rows)
3. Replacing the missing values with mean\median\mode- The choice should be made after checking the central tendency and the skewness. For example use mean only when the data is not skewed or else use median.
4. By predictive algorithms by taking the rest of the non-missing values as features and missing value as the target.
5. Assigning a new category value for NAN Values. It preserves the variance, but might give high random data when missing values are in large quantity.
6. By using unsupervised algorithms like k-mean clustering

#install.packages("mice") library(mice) mice.impute.logreg(d2$Married, d2$LoanAmount)

Assigning a new category value for NAN Values in categorical columns for future analysis

[56]:
```
d2$Gender[is.na(d2$Gender)] = "unkown"
d2$Married[is.na(d2$Married)] = "unkown"
d2$Dependents[is.na(d2$Dependents)] = "unkown"
d2$Self_Employed[is.na(d2$Self_Employed)] = "unkown"
d2$Credit_History[is.na(d2$Credit_History)] = "unkown"
```

For numerical variables we compute missing value with median here as we cannot use mean for skewed variables

[73]:
```
#install.packages('moments')
library(moments)
```

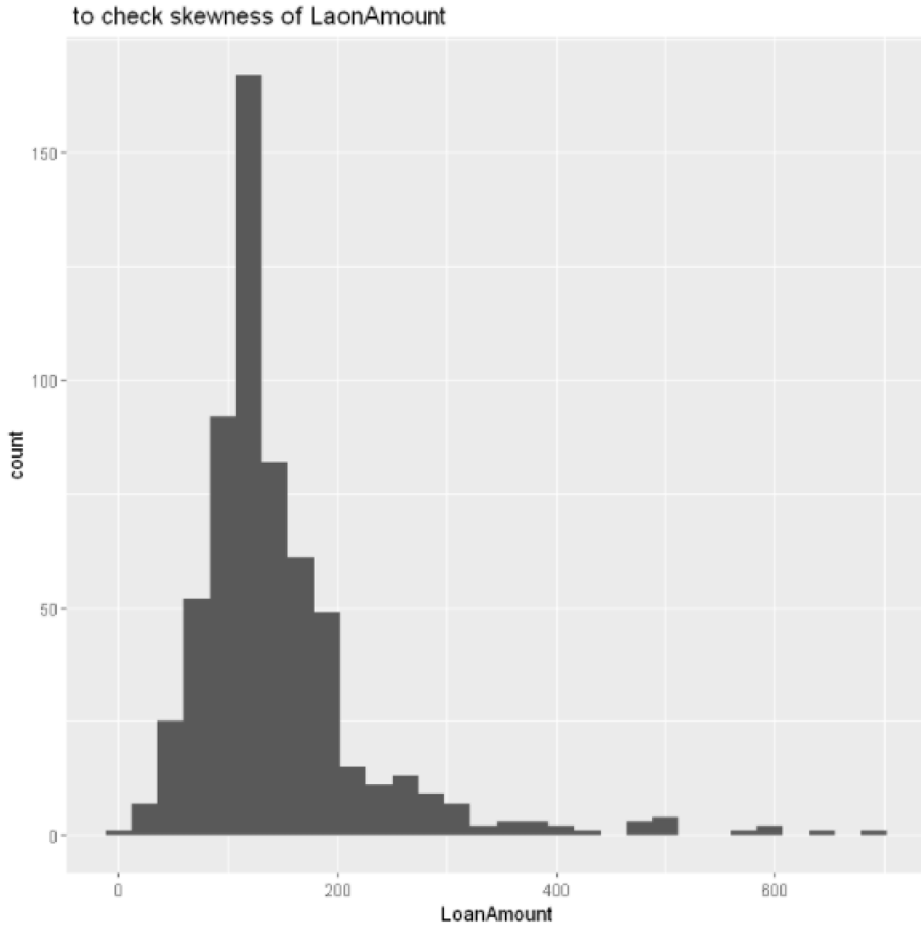
[74]:
```
print(skewness(d2$LoanAmount))
```

```
[1] 2.736347
```

[75]:
```
print(skewness(d2$Loan_Amount_Term))
```

```
[1] -2.39624
```

In [76]:
```
ggplot(data=d2, aes(LoanAmount)) + geom_histogram(bins = 30, position = position_dodge())+xlab("LoanAmount")+ ggtitle(" to c
```



As stated loan Amount is left skewed hence mean imputation would not be right

In [61]:
```
d2$LoanAmount[is.na(d2$LoanAmount)] <- median(d2$LoanAmount, na.rm = T)
d2$Loan_Amount_Term[is.na(d2$Loan_Amount_Term)] <- median(d2$Loan_Amount_Term, na.rm = T)
```

In [62]:
```
colSums(is.na(d2))
```

...1	0
Loan_ID	0
Gender	0
Married	0
Dependents	0
Education	0
Self_Employed	0
ApplicantIncome	0
CoapplicantIncome	0
LoanAmount	0
Loan_Amount_Term	0
Credit_History	0
Property_Area	0
Loan_Status	0

6. (12) Create the variable "aggregatedInocme" for each loan application.

6. Creating the variable aggregatedInocme

63]:
```
d2$aggregatedIncome <- d2$ApplicantIncome + d2$CoapplicantIncome
head(d2)
```

ducation	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status	aggregatedIncome
Graduate	No	5849	0	128	360	1	Urban	Y	5849
Graduate	No	4583	1508	128	360	1	Rural	N	6091
Graduate	Yes	3000	0	66	360	1	Urban	Y	3000
Not Graduate	No	2583	2358	120	360	1	Urban	Y	4941
Graduate	No	6000	0	141	360	1	Urban	Y	6000
Graduate	Yes	5417	4196	267	360	1	Urban	Y	9613

7. (12) Define and create your own three categories of "aggregatedIncome": high, medium, and low.

7. three categories of aggregatedIncome--> high, medium, and low.

4]:
```
d2 <- d2 %>% mutate(Group =
                      case_when(aggregatedIncome <= 5000 ~ "low",
                                aggregatedIncome <= 8000 ~ "medium",
                                aggregatedIncome >= 8001 ~ "high"))
```

5]:
```
head(d2,10)
```

Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status	aggregatedIncome	Group
No	5849	0	128	360	1	Urban	Y	5849	medium
No	4583	1508	128	360	1	Rural	N	6091	medium
Yes	3000	0	66	360	1	Urban	Y	3000	low
No	2583	2358	120	360	1	Urban	Y	4941	low
No	6000	0	141	360	1	Urban	Y	6000	medium
Yes	5417	4196	267	360	1	Urban	Y	9613	high
No	2333	1516	95	360	1	Urban	Y	3849	low
No	3036	2504	158	360	0	Semiurban	N	5540	medium
No	4006	1526	168	360	1	Urban	Y	5532	medium
No	12841	10968	349	360	1	Semiurban	N	23809	high

8. (12) In each your defined categories of "aggregatedIncome", what percentage of applications received their loan approvals?

8. Based on category percentage of applications received their loan approvals

6]:
```
low_percent <- sum(d2$Group=="low" & d2$Loan_Status == "Y")/sum(d2$Group=="low")*100
low_percent
```

67.037037037037

7]:
```
medium_percent <- sum(d2$Group=="medium" & d2$Loan_Status == "Y")/sum(d2$Group=="medium")*100
medium_percent
```

71.2264150943396

8]:
```
high_percent <- sum(d2$Group=="high" & d2$Loan_Status == "Y")/sum(d2$Group=="high")*100
high_percent
```

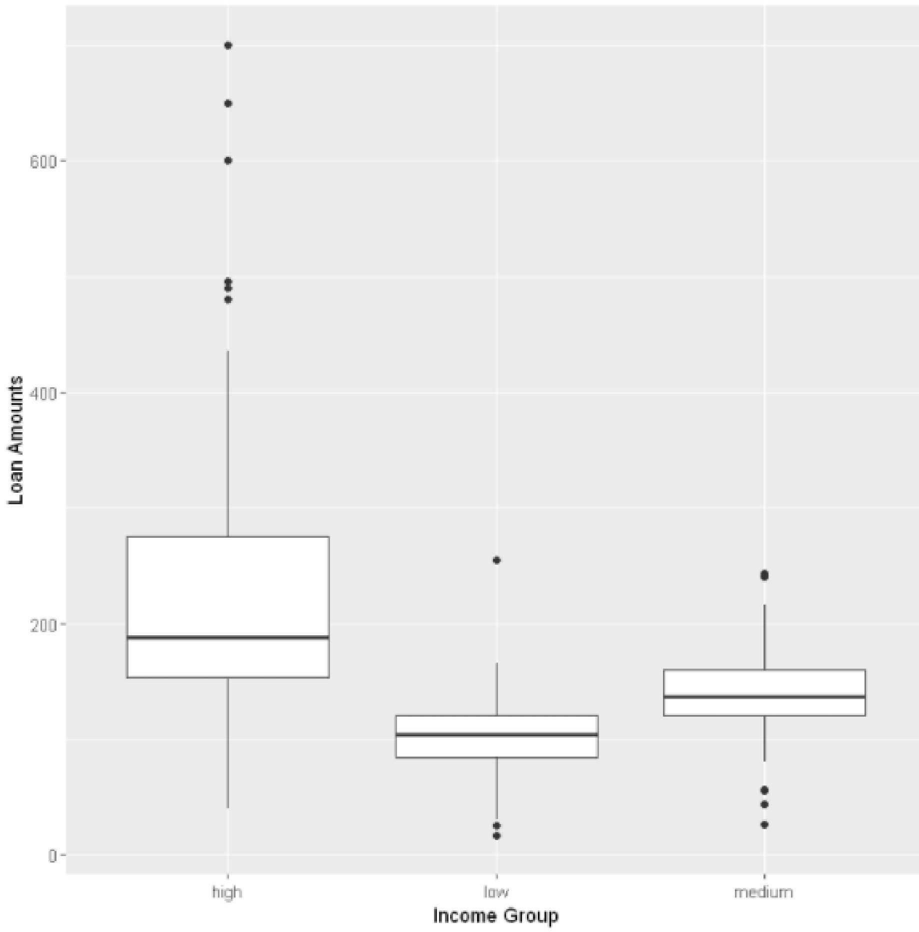
68.1818181818182

We can observe that the medium percent has highest percentage of loan approvals

9. (12) Comparing with loan sizes, will you conclude any insight?

9. Conclusion based on loan sizes

```
sub <- subset(d2,Loan_Status=="Y")
ggplot(sub, aes(y=LoanAmount, x=Group)) + geom_boxplot() + xlab("Income Group") + ylab("Loan Amounts")
```



As we selected only those columns where the loans were approved, we can see the distribution of loan amount approval for the 3 different income categories.

From this we can conclude that Higher income category more is the sanctioned loan amount. That does hold true as more a customers income more the bank can trust in loaning a larger amount to the customer.