Contents

1 Team : We Showed Up

1.1 Team Members :

- Nitya Patel (202051129)
- Prathak Garg (202051144)
- Pratyush Agrawal(202051145)
- Prince Rakholiya (202051147)

1.2 Libraries Used :

```
library(bnlearn)     # for bayesian network
library(bnclassify) # application based algorithms for bayesian network classifiers (like predictions,a
library(dplyr)       # data manipulation (similar to pandas in python)
library(ggplot2)     # creating the plots (similar to matplotlib in python)
library(graph)       # creates graphical models
```

1.3 Problem 1

Consider grades earned in each of the courses as random variables and learn the dependencies between courses.

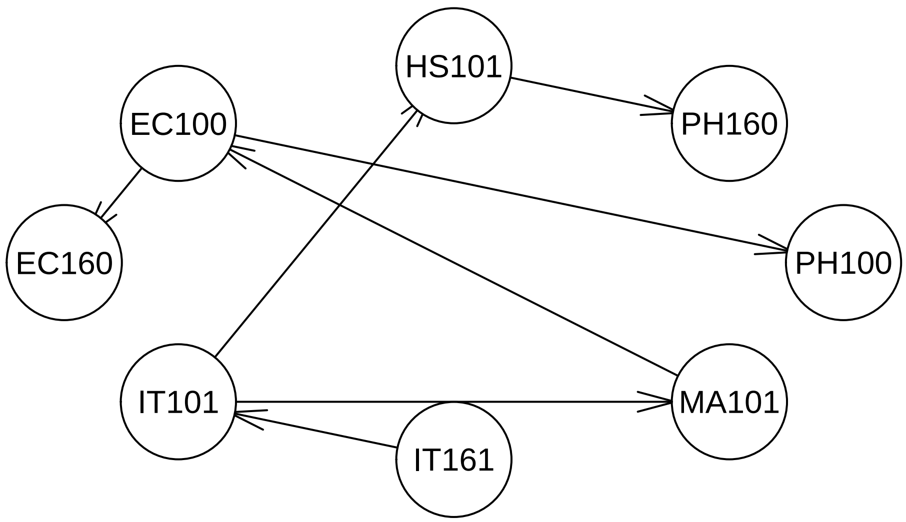
First we read the data into R.

```
course.grades<-read.table("2020_bn_nb_data.txt",head=TRUE)
head(course.grades)
```

```
##   EC100 EC160 IT101 IT161 MA101 PH100 PH160 HS101 QP
## 1    BC    CC    BB    BC    CC    BC    AA    BB  y
## 2    CC    BC    BB    BB    CC    BC    AB    BB  y
## 3    AB    BB    AB    AB    BB    CC    BC    AB  y
## 4    BC    CC    BB    BB    BB    BB    BC    BB  y
## 5    BC    AB    CD    BC    BC    BC    BC    CD  y
## 6    DD    CC    DD    CD    CD    CC    BC    BC  n
```

We now get dependencies between the courses.

```
course.grades<- lapply(course.grades,as.factor)
course.grades<- data.frame(course.grades)
course.grades.net<- hc(course.grades[,-9], score='k2')
plot(course.grades.net)
```



1.4 Problem 2

Using the data, learn the CPTs for each course node.

The first argument, "course.grades.net", is a pre-specified network structure or graphical model that defines the relationships between variables in the dataset. The second argument, "course.grades[,-9]", is the **data** used to estimate the parameters of the model, excluding the 9th column.

```
course.grades.fit <- bn.fit(course.grades.net,course.grades[,-9])
course.grades.fit
```

```
##
##   Bayesian network parameters
##
##   Parameters of node EC100 (multinomial distribution)
##
## Conditional probability table:
```

```
##
##      MA101
## EC100         AA         AB         BB         BC         CC         CD
##    AA 0.75000000 0.07692308 0.03846154 0.01851852 0.00000000 0.00000000
##    AB 0.00000000 0.46153846 0.25000000 0.05555556 0.00000000 0.00000000
##    BB 0.25000000 0.23076923 0.32692308 0.22222222 0.04081633 0.00000000
##    BC 0.00000000 0.15384615 0.28846154 0.27777778 0.32653061 0.00000000
##    CC 0.00000000 0.07692308 0.09615385 0.24074074 0.32653061 0.04166667
##    CD 0.00000000 0.00000000 0.00000000 0.12962963 0.26530612 0.33333333
##    DD 0.00000000 0.00000000 0.00000000 0.03703704 0.04081633 0.50000000
##    F  0.00000000 0.00000000 0.00000000 0.01851852 0.00000000 0.12500000
##      MA101
## EC100         DD          F
##    AA 0.00000000 0.00000000
##    AB 0.00000000 0.00000000
##    BB 0.00000000 0.00000000
##    BC 0.00000000 0.00000000
##    CC 0.00000000 0.00000000
##    CD 0.04761905 0.00000000
##    DD 0.19047619 0.00000000
##    F  0.76190476 1.00000000
##
##   Parameters of node EC160 (multinomial distribution)
##
## Conditional probability table:
##
##      EC100
## EC160         AA         AB         BB         BC         CC         CD
##    AA 0.42857143 0.22727273 0.05714286 0.04166667 0.00000000 0.00000000
##    AB 0.42857143 0.22727273 0.08571429 0.04166667 0.08333333 0.00000000
##    BB 0.14285714 0.31818182 0.20000000 0.22916667 0.08333333 0.03448276
##    BC 0.00000000 0.22727273 0.42857143 0.43750000 0.36111111 0.17241379
##    CC 0.00000000 0.00000000 0.22857143 0.25000000 0.30555556 0.34482759
##    CD 0.00000000 0.00000000 0.00000000 0.00000000 0.11111111 0.27586207
##    DD 0.00000000 0.00000000 0.00000000 0.00000000 0.05555556 0.17241379
##    F  0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##      EC100
## EC160         DD          F
##    AA 0.00000000 0.00000000
##    AB 0.00000000 0.00000000
##    BB 0.05000000 0.00000000
##    BC 0.00000000 0.00000000
##    CC 0.25000000 0.02857143
##    CD 0.55000000 0.40000000
##    DD 0.15000000 0.34285714
##    F  0.00000000 0.22857143
##
##   Parameters of node IT101 (multinomial distribution)
##
## Conditional probability table:
##
##      IT161
## IT101         AA         AB         BB         BC         CC         CD
##    AA 0.35000000 0.08000000 0.05714286 0.02040816 0.00000000 0.00000000
```

```
##    AB 0.30000000 0.40000000 0.17142857 0.02040816 0.02380952 0.02857143
##    BB 0.25000000 0.40000000 0.31428571 0.14285714 0.00000000 0.02857143
##    BC 0.10000000 0.04000000 0.28571429 0.36734694 0.28571429 0.14285714
##    CC 0.00000000 0.08000000 0.14285714 0.32653061 0.33333333 0.11428571
##    CD 0.00000000 0.00000000 0.02857143 0.12244898 0.26190476 0.31428571
##    DD 0.00000000 0.00000000 0.00000000 0.00000000 0.04761905 0.34285714
##    F  0.00000000 0.00000000 0.00000000 0.00000000 0.04761905 0.02857143
##      IT161
## IT101         DD          F
##    AA 0.00000000 0.00000000
##    AB 0.00000000 0.00000000
##    BB 0.00000000 0.00000000
##    BC 0.04347826 0.00000000
##    CC 0.04347826 0.00000000
##    CD 0.21739130 0.33333333
##    DD 0.39130435 0.00000000
##    F  0.30434783 0.66666667
##
##   Parameters of node IT161 (multinomial distribution)
##
## Conditional probability table:
##          AA         AB         BB         BC         CC         CD         DD
## 0.08620690 0.10775862 0.15086207 0.21120690 0.18103448 0.15086207 0.09913793
##          F
## 0.01293103
##
##   Parameters of node MA101 (multinomial distribution)
##
## Conditional probability table:
##
##      IT101
## MA101         AA         AB         BB         BC         CC         CD
##    AA 0.16666667 0.04000000 0.00000000 0.00000000 0.02380952 0.00000000
##    AB 0.25000000 0.20000000 0.02941176 0.08163265 0.00000000 0.00000000
##    BB 0.33333333 0.56000000 0.38235294 0.22448980 0.19047619 0.05714286
##    BC 0.16666667 0.16000000 0.29411765 0.36734694 0.23809524 0.22857143
##    CC 0.08333333 0.00000000 0.20588235 0.28571429 0.35714286 0.31428571
##    CD 0.00000000 0.04000000 0.08823529 0.02040816 0.16666667 0.11428571
##    DD 0.00000000 0.00000000 0.00000000 0.02040816 0.02380952 0.22857143
##    F  0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.05714286
##      IT101
## MA101         DD          F
##    AA 0.00000000 0.00000000
##    AB 0.00000000 0.00000000
##    BB 0.00000000 0.00000000
##    BC 0.08695652 0.00000000
##    CC 0.04347826 0.00000000
##    CD 0.30434783 0.08333333
##    DD 0.39130435 0.16666667
##    F  0.17391304 0.75000000
##
##   Parameters of node PH100 (multinomial distribution)
##
## Conditional probability table:
```

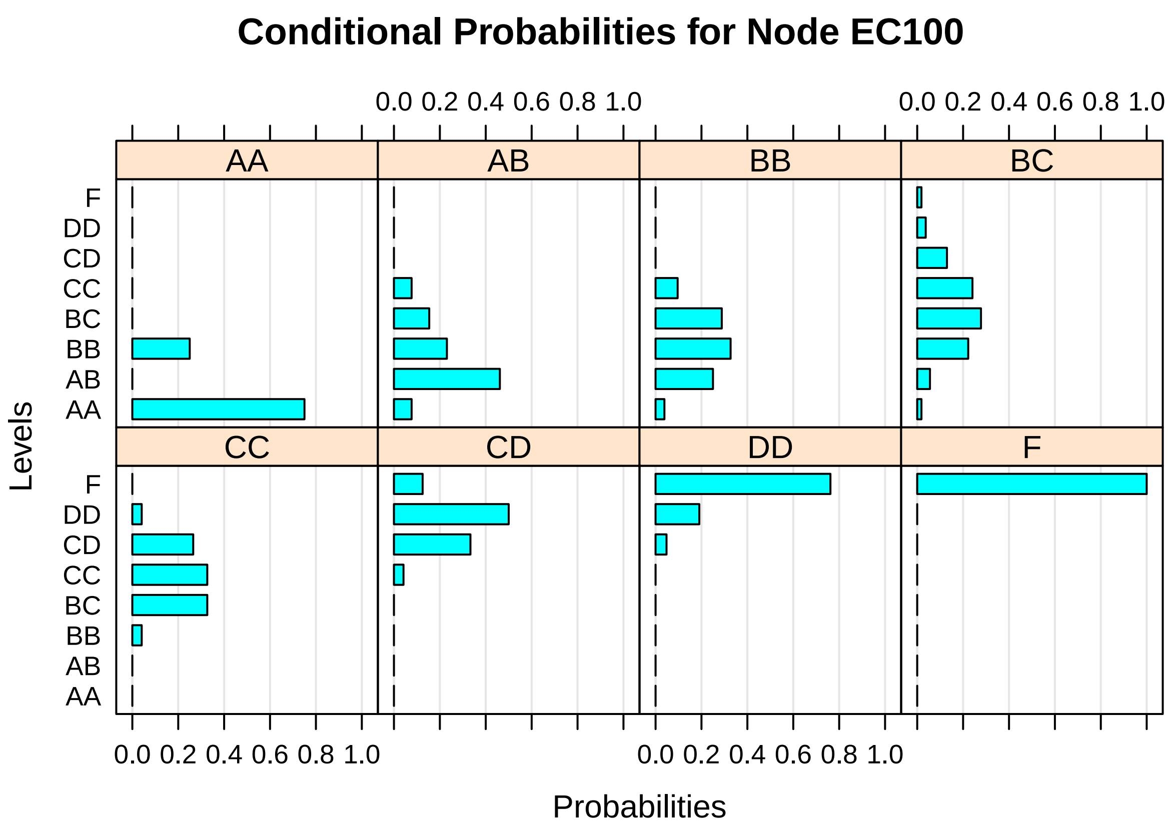
```
##
##      EC100
## PH100         AA         AB         BB         BC         CC         CD
##    AA 0.71428571 0.40909091 0.22857143 0.08333333 0.00000000 0.00000000
##    AB 0.14285714 0.31818182 0.20000000 0.18750000 0.05555556 0.00000000
##    BB 0.00000000 0.18181818 0.31428571 0.29166667 0.13888889 0.03448276
##    BC 0.14285714 0.04545455 0.14285714 0.22916667 0.33333333 0.13793103
##    CC 0.00000000 0.04545455 0.11428571 0.18750000 0.25000000 0.41379310
##    CD 0.00000000 0.00000000 0.00000000 0.02083333 0.19444444 0.31034483
##    DD 0.00000000 0.00000000 0.00000000 0.00000000 0.02777778 0.10344828
##    F  0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##      EC100
## PH100         DD          F
##    AA 0.00000000 0.00000000
##    AB 0.00000000 0.00000000
##    BB 0.05000000 0.00000000
##    BC 0.00000000 0.00000000
##    CC 0.20000000 0.02857143
##    CD 0.45000000 0.11428571
##    DD 0.20000000 0.45714286
##    F  0.10000000 0.40000000
##
##   Parameters of node PH160 (multinomial distribution)
##
## Conditional probability table:
##
##      HS101
## PH160         AA         AB         BB         BC         CC         CD
##    AA 0.23809524 0.17647059 0.05000000 0.11111111 0.07692308 0.10000000
##    AB 0.23809524 0.11764706 0.15000000 0.13888889 0.07692308 0.10000000
##    BB 0.16666667 0.26470588 0.17500000 0.16666667 0.00000000 0.00000000
##    BC 0.21428571 0.32352941 0.45000000 0.22222222 0.50000000 0.30000000
##    CC 0.09523810 0.08823529 0.12500000 0.30555556 0.15384615 0.45000000
##    CD 0.04761905 0.02941176 0.02500000 0.05555556 0.11538462 0.05000000
##    DD 0.00000000 0.00000000 0.02500000 0.00000000 0.07692308 0.00000000
##    F  0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##      HS101
## PH160         DD          F
##    AA 0.03448276 0.00000000
##    AB 0.10344828 0.00000000
##    BB 0.00000000 0.20000000
##    BC 0.10344828 0.00000000
##    CC 0.24137931 0.00000000
##    CD 0.37931034 0.00000000
##    DD 0.13793103 0.40000000
##    F  0.00000000 0.40000000
##
##   Parameters of node HS101 (multinomial distribution)
##
## Conditional probability table:
##
##      IT101
## HS101         AA         AB         BB         BC         CC         CD
##    AA 0.58333333 0.56000000 0.32352941 0.10204082 0.07142857 0.05714286
```

```
##    AB 0.33333333 0.24000000 0.11764706 0.22448980 0.14285714 0.08571429
##    BB 0.00000000 0.12000000 0.26470588 0.26530612 0.26190476 0.11428571
##    BC 0.08333333 0.08000000 0.08823529 0.24489796 0.23809524 0.20000000
##    CC 0.00000000 0.00000000 0.11764706 0.12244898 0.14285714 0.11428571
##    CD 0.00000000 0.00000000 0.05882353 0.02040816 0.14285714 0.20000000
##    DD 0.00000000 0.00000000 0.02941176 0.02040816 0.00000000 0.22857143
##    F  0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##      IT101
## HS101         DD          F
##    AA 0.00000000 0.00000000
##    AB 0.00000000 0.00000000
##    BB 0.00000000 0.00000000
##    BC 0.04347826 0.00000000
##    CC 0.26086957 0.00000000
##    CD 0.13043478 0.08333333
##    DD 0.52173913 0.58333333
##    F  0.04347826 0.33333333
```

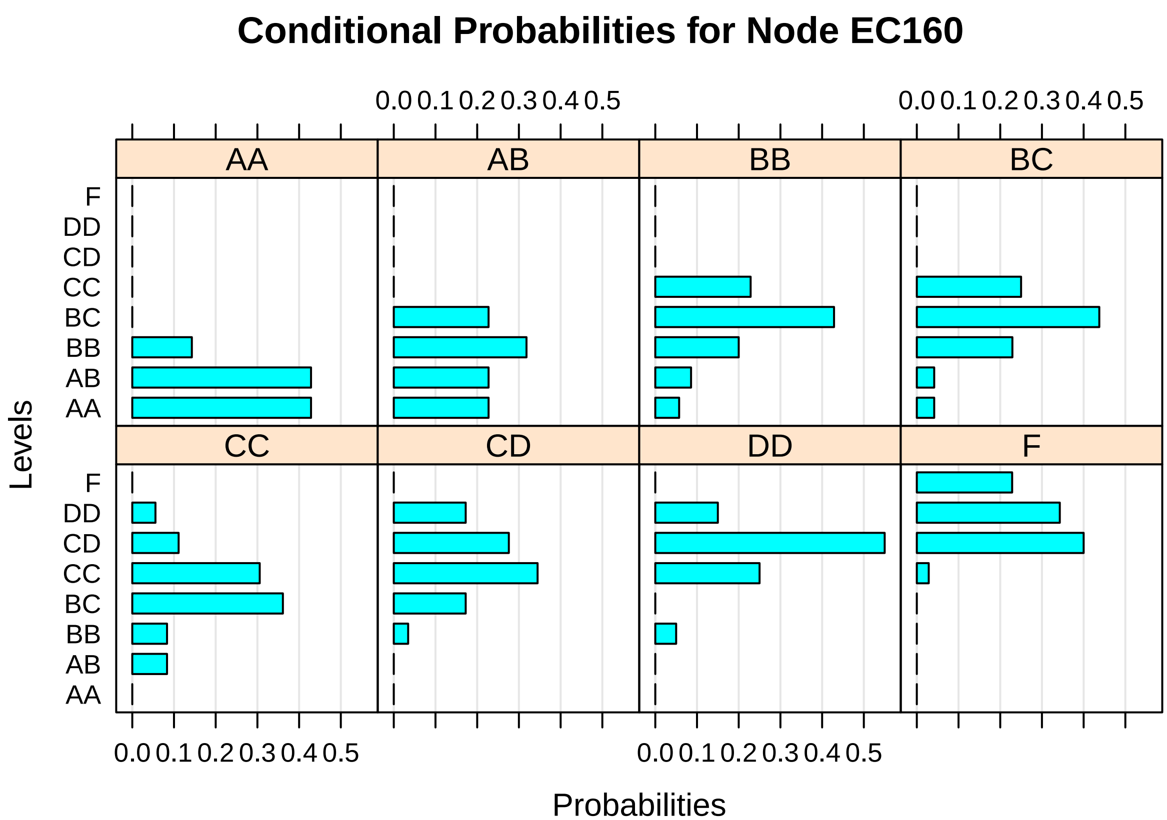
Let us visualise these a little bit.

For each of the variables in the model, the corresponding CPT is passed as an argument to the "bn.fit.barchart" function, creating a bar chart that visually represents the **probability distribution** of the variable given its parents in the network.

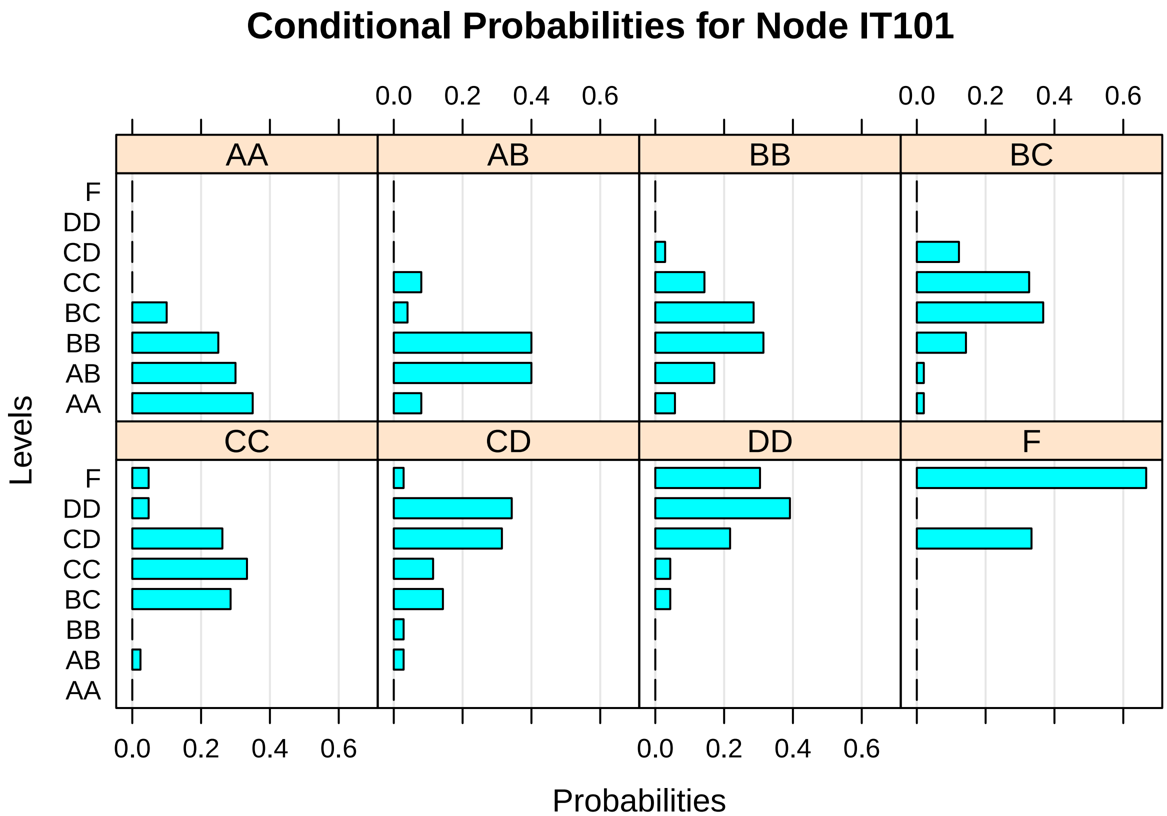
```
bn.fit.barchart(course.grades.fit$EC100)
```



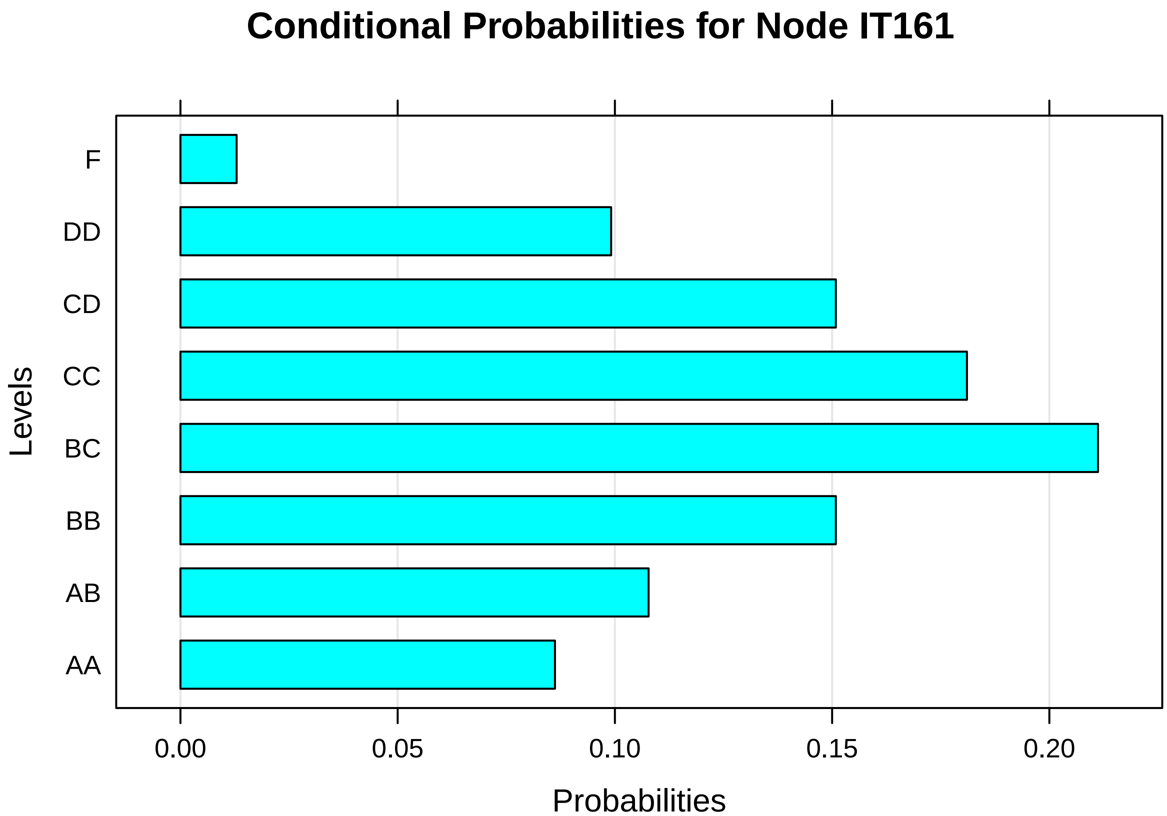
```
bn.fit.barchart(course.grades.fit$EC160)
```



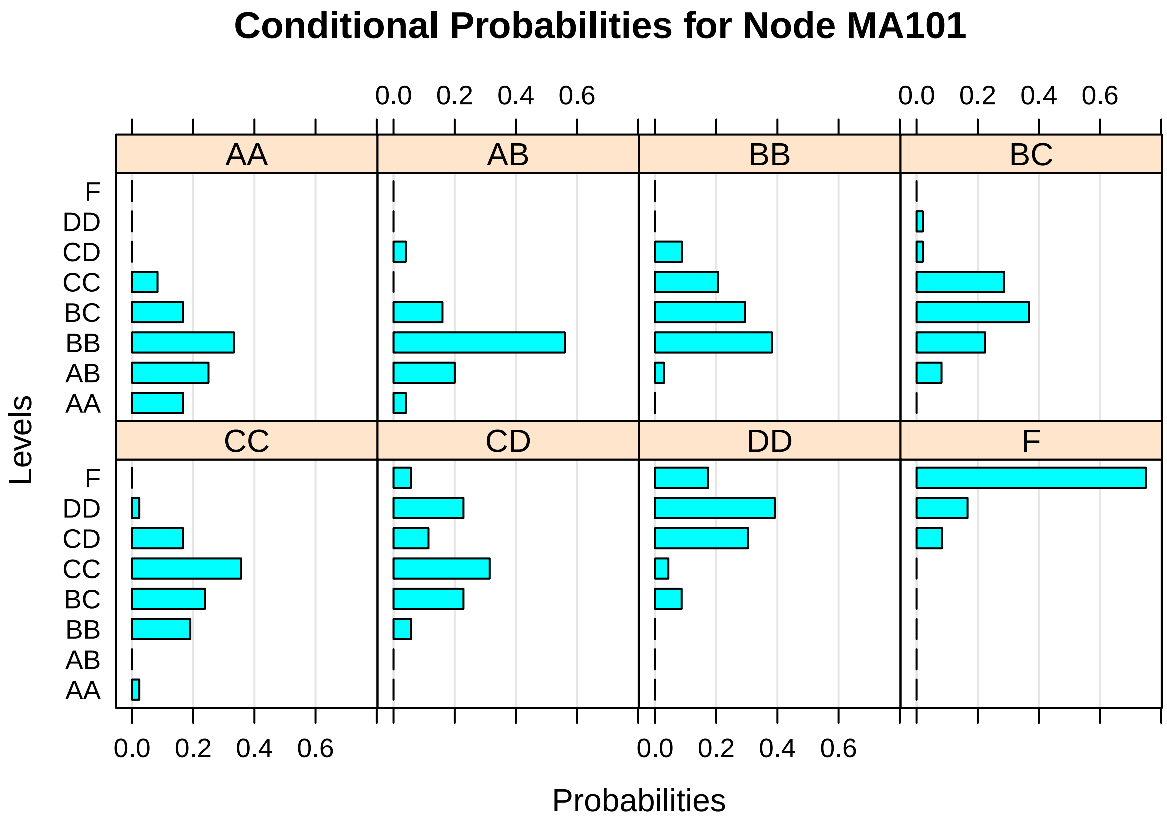

```
bn.fit.barchart(course.grades.fit$IT101)
```



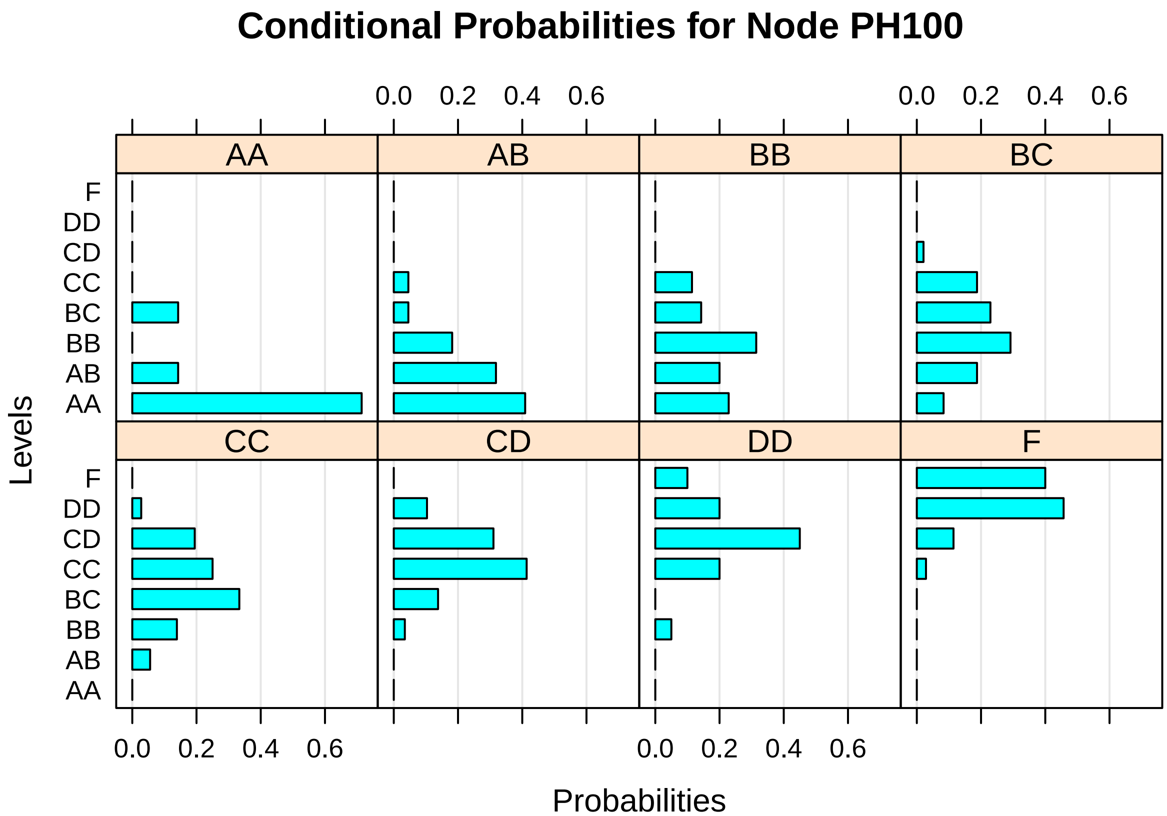

```
bn.fit.barchart(course.grades.fit$IT161)
```



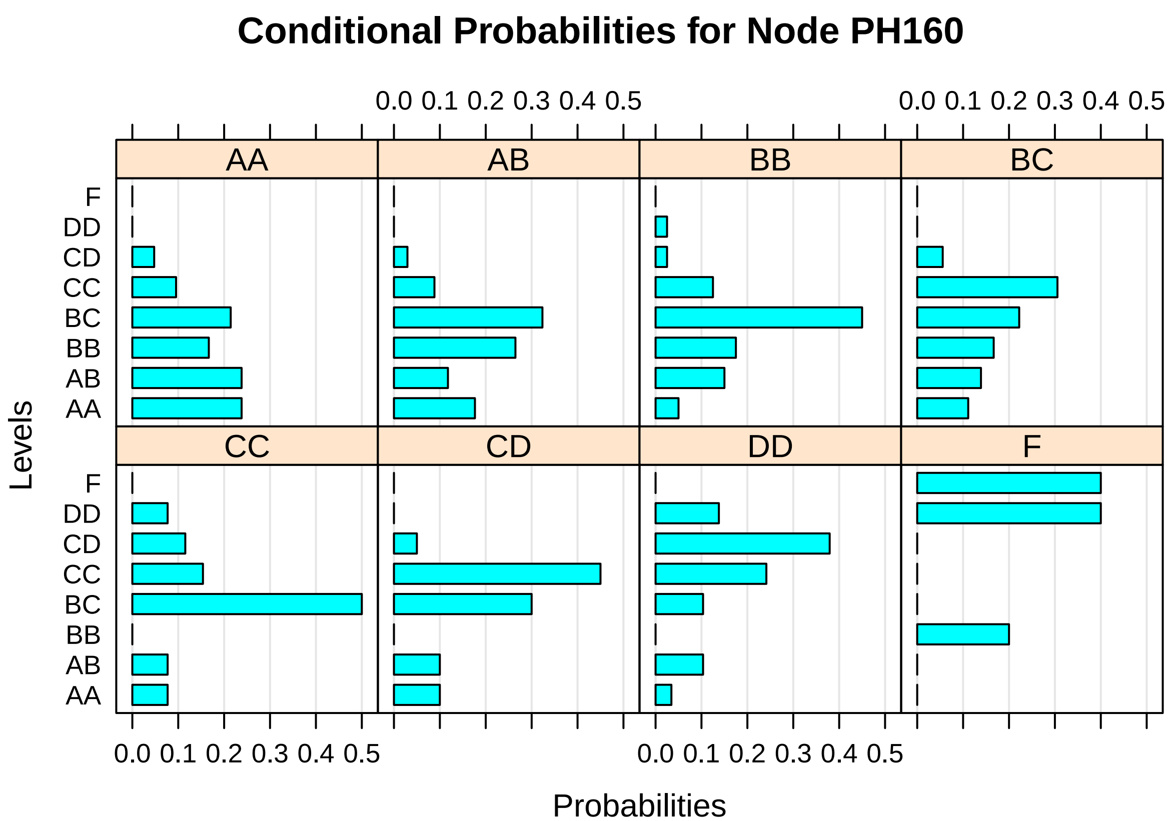

```
bn.fit.barchart(course.grades.fit$MA101)
```




```
bn.fit.barchart(course.grades.fit$PH100)
```




```
bn.fit.barchart(course.grades.fit$PH160)
```



1.5 Problem 3

What grade will a student get in PH100 if he earns DD in EC100, CC in IT101 and CD in MA101.

It creates a data frame containing the **marginal probability distribution** of "PH100" given specific values for its parent variables "EC100", "IT101", and "MA101". The **cpdist** function from the bnlearn library is used to calculate the marginal probabilities based on the fitted model. The "nodes" argument specifies the target node, which is "PH100", and the "evidence" argument specifies the values of its parent variables.

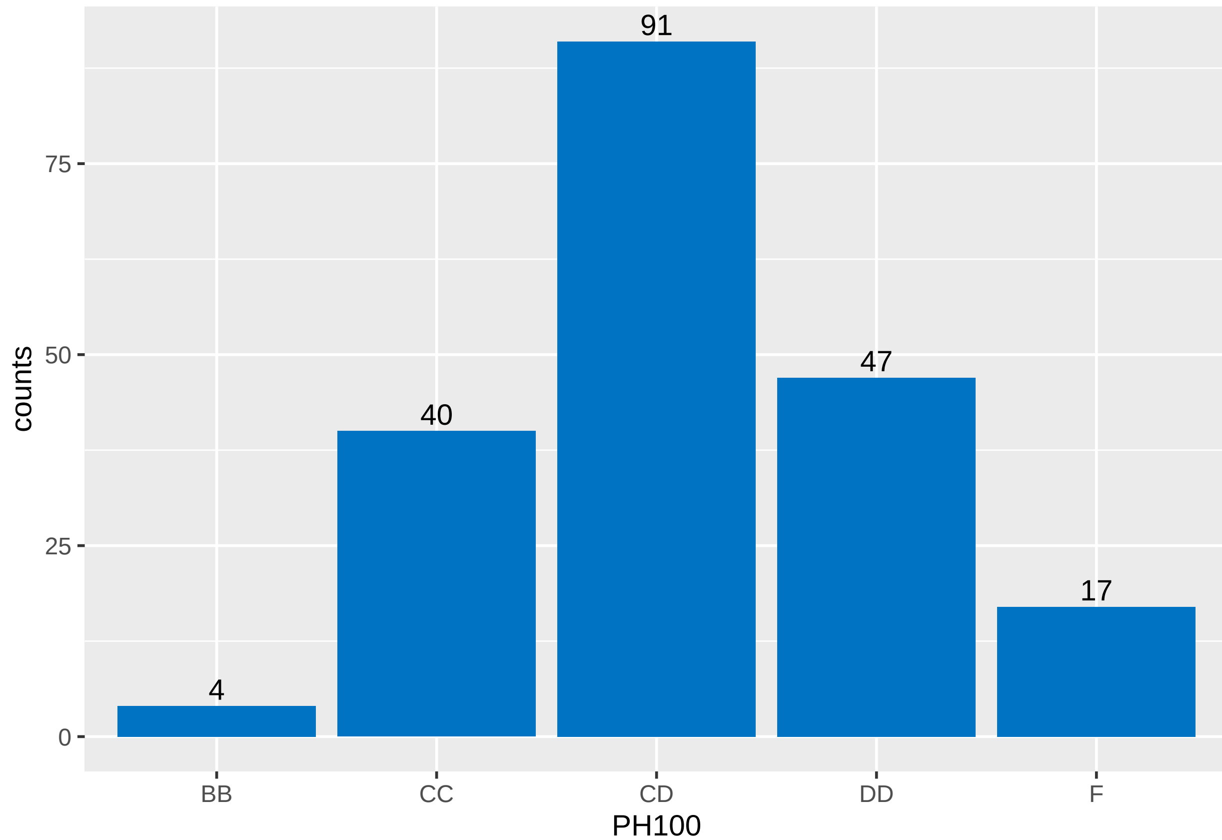
Next we are using the dplyr library to group the data frame "course.grades.PH100" by the "PH100" variable and to summarize the counts of each value of "PH100". The resulting data frame is stored in the object "df".

```
course.grades.PH100 <- data.frame( cpdist(course.grades.fit, nodes = c("PH100"),evidence = ( (EC100=="DI

df <- course.grades.PH100 %>%
  group_by(PH100) %>%
  summarise(counts = n())
```

Here we are using the ggplot2 library to create a bar plot of the summarized data. The ggplot function takes the "df" data frame as its data argument, and the "aes" function specifies that the x-axis should represent the "PH100" variable and the y-axis should represent the "counts". The "geom_bar" function creates the bar plot with a blue fill color, and the "geom_text" function adds the count values as labels to the bars. The "vjust" argument adjusts the vertical position of the labels to avoid overlapping with the bars.

```
ggplot(df, aes(x = PH100, y = counts)) +
  geom_bar(fill = "#0073C2FF", stat = "identity") +
  geom_text(aes(label = counts), vjust = -0.3)
```



1.6 Problem 4

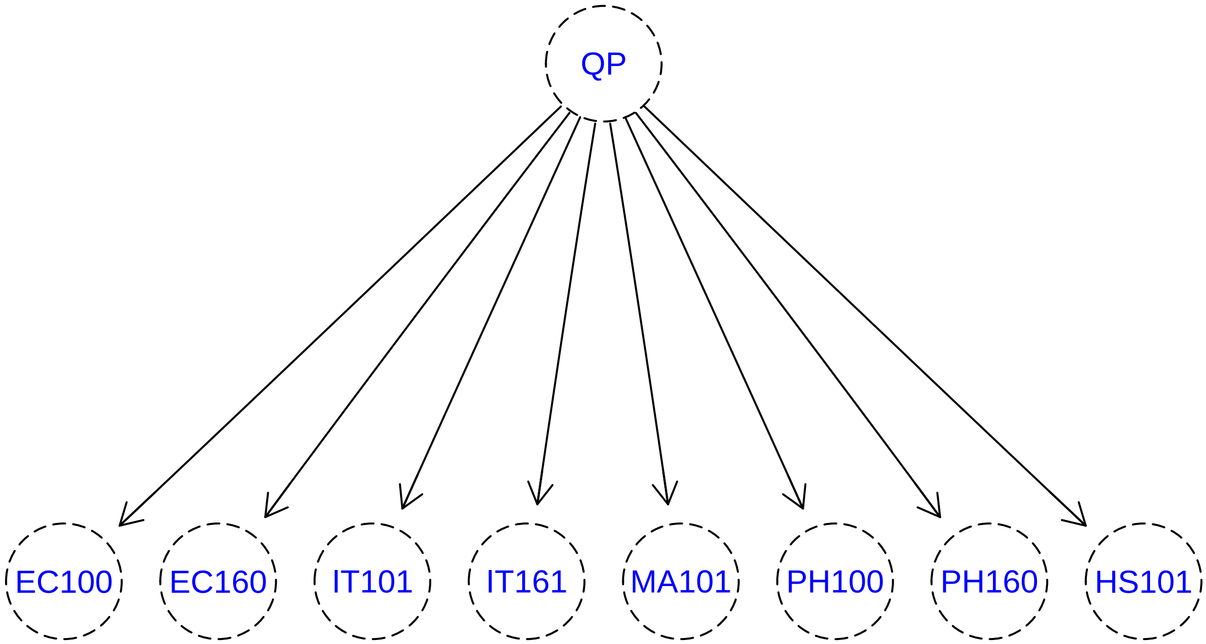
The last column in the data file indicates whether a student qualifies for an internship program or not. From the given data, take 70 percent data for training, and build a naive Bayes classifier (considering that the grades earned in different courses are independent of each other) which takes in the student's performance and returns the qualification status with a probability. Test your classifier on the remaining 30 percent data.

Let us split the data first.

```
set.seed(101)
sample <-  sample.int(n = nrow(course.grades), size = floor(0.7*nrow(course.grades)), replace=F)
course.grades.train <-course.grades[sample,]
course.grades.test<- course.grades[-sample,]
```

Let us create a classifier based of Bayesian System.

```
nb.grades <- nb(class="QP", dataset=course.grades.train)
plot(nb.grades)
```



Training the classifier.

```
nb.grades <- lp(nb.grades,course.grades.train, smooth=0)
nb.grades$.params
```

```
## $EC100
##      QP
## EC100           n           y
##    AA 0.000000000 0.037037037
##    AB 0.000000000 0.148148148
##    BB 0.000000000 0.222222222
##    BC 0.000000000 0.287037037
##    CC 0.055555556 0.185185185
##    CD 0.166666667 0.111111111
##    DD 0.314814815 0.009259259
##    F  0.462962963 0.000000000
##
## $EC160
##      QP
## EC160          n          y
##    AA 0.00000000 0.07407407
##    AB 0.00000000 0.09259259
##    BB 0.01851852 0.19444444
##    BC 0.01851852 0.36111111
##    CC 0.12962963 0.22222222
##    CD 0.46296296 0.03703704
```

```
##    DD 0.25925926 0.01851852
##    F  0.11111111 0.00000000
##
## $IT101
##      QP
## IT101          n          y
##    AA 0.00000000 0.06481481
##    AB 0.00000000 0.13888889
##    BB 0.05555556 0.23148148
##    BC 0.01851852 0.28703704
##    CC 0.11111111 0.20370370
##    CD 0.33333333 0.07407407
##    DD 0.35185185 0.00000000
##    F  0.12962963 0.00000000
##
## $IT161
##      QP
## IT161           n           y
##    AA 0.000000000 0.111111111
##    AB 0.018518519 0.148148148
##    BB 0.018518519 0.203703704
##    BC 0.018518519 0.305555556
##    CC 0.185185185 0.175925926
##    CD 0.444444444 0.046296296
##    DD 0.259259259 0.009259259
##    F  0.055555556 0.000000000
##
## $MA101
##      QP
## MA101          n          y
##    AA 0.00000000 0.02777778
##    AB 0.00000000 0.08333333
##    BB 0.00000000 0.32407407
##    BC 0.09259259 0.27777778
##    CC 0.16666667 0.26851852
##    CD 0.25925926 0.01851852
##    DD 0.31481481 0.00000000
##    F  0.16666667 0.00000000
##
## $PH100
##      QP
## PH100          n          y
##    AA 0.00000000 0.14814815
##    AB 0.00000000 0.17592593
##    BB 0.01851852 0.21296296
##    BC 0.03703704 0.23148148
##    CC 0.14814815 0.17592593
##    CD 0.20370370 0.05555556
##    DD 0.38888889 0.00000000
##    F  0.20370370 0.00000000
##
## $PH160
##      QP
## PH160          n          y
```

```
##    AA 0.05555556 0.16666667
##    AB 0.11111111 0.15740741
##    BB 0.01851852 0.16666667
##    BC 0.16666667 0.34259259
##    CC 0.31481481 0.12962963
##    CD 0.16666667 0.03703704
##    DD 0.12962963 0.00000000
##    F  0.03703704 0.00000000
##
## $HS101
##      QP
## HS101           n           y
##    AA 0.000000000 0.231481481
##    AB 0.000000000 0.212962963
##    BB 0.055555556 0.250000000
##    BC 0.148148148 0.157407407
##    CC 0.148148148 0.092592593
##    CD 0.203703704 0.046296296
##    DD 0.370370370 0.009259259
##    F  0.074074074 0.000000000
##
## $QP
## QP
##         n         y
## 0.3333333 0.6666667
```

Testing the classifier.

```
p <- predict(nb.grades,course.grades.test)
cm <- table(predicted_on_test_data=p, actual_data=course.grades.test$QP)
cm
```

```
##                         actual_data
## predicted_on_test_data  n  y
##                       n 17  0
##                       y  1 52
```

Evaluating results.

```
bnclassify:::accuracy(p, course.grades.test$QP)
```

```
## [1] 0.9857143
```

1.7 References :

- https://github.com/TanmayAmbadkar/CS302-AI