Homework 2

1. Generate a tree graph that represents flipping a coin 4 times, let A be the event “the first outcome is tails” , B the event “ the second outcome is head” and C the event “the third outcome is tails” calculate

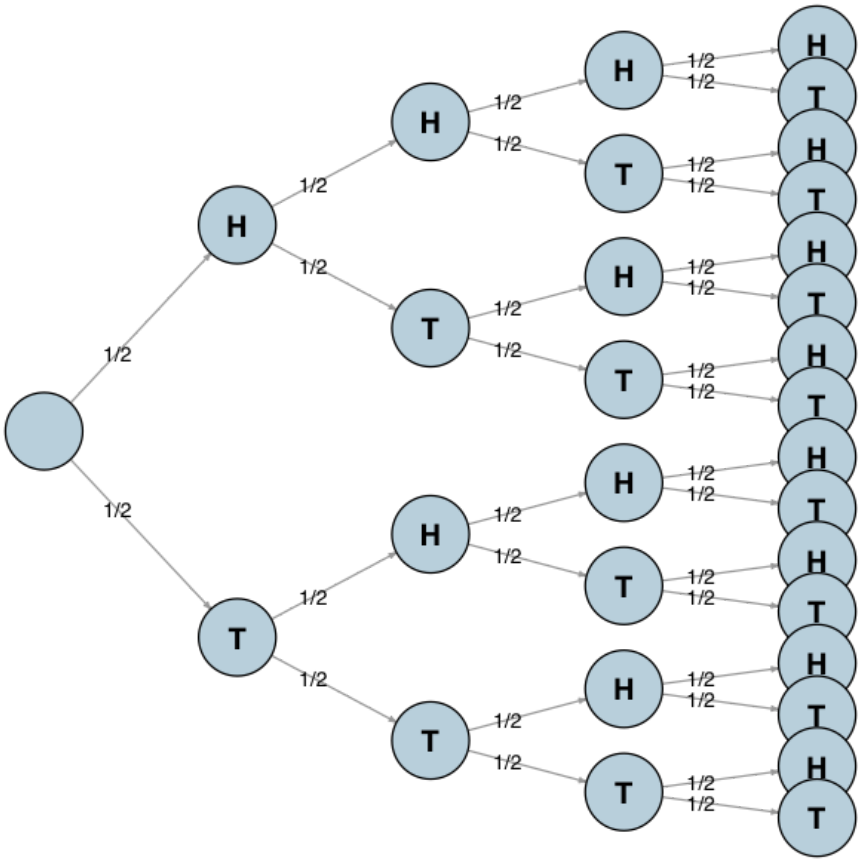
p(AUBUC) = P (A U B U C) = P(A) + P(B) + P(C) - P(A ∩ B) - P(A ∩ C) - P(B ∩ C) + P(A ∩ B ∩ C)

In [14]:
```
library(igraph)
g <- graph.tree(n = 2^5 - 1, children = 2)
n_l = c("H","T")
node_labels <- c("",replicate(15,n_l))
edge_labels <- c("1/2")
edge_label2 = replicate(30,edge_labels)
```

In [15]:
```
V(g)$color <- "#C4D8E2"
#V(g)$color[3] <- "white"
#V(g)$color[4] <- "green"

#assign position
coords <- layout_(g, as_tree())
coord2 = matrix(c(-coords[,2],-coords[,1]),ncol = 2)
```

In [16]:
```
plot(g,
     layout = coord2,           # draw graph as tree
     vertex.size = 20,                # node size
     vertex.color = V(g)$color,        # node color
     vertex.label = node_labels,      # node labels
     vertex.label.cex = 1,           # node label size
     vertex.label.family = "Helvetica", # node label family
     vertex.label.font = 2,           # node label type (bold)
     vertex.label.color = '#000000',    # node label size
     edge.label = edge_label2,         # edge labels
     edge.label.cex = .7,              # edge label size
     edge.label.family = "Helvetica",   # edge label family
     edge.label.font = 1,              # edge label font type (bold)
     edge.label.color = '#000000',      # edge label color
     edge.arrow.size = 0.2,            # arrow size
     edge.arrow.width = 1             # arrow width
)
```



In [ ]:
```
p(AUBUC) = P (A U B U C) = P(A) + P(B) + P(C) - P(A ∩ B) - P(A ∩ C) - P(B ∩ C) + P(A ∩ B
 ∩ C)
=1/2+ 1/2 + 1/2 - 4/16-4/16-4/16+2/16 = 0.875
```

2. From the Dataset Diabetes, construct contingency tables for the following variable combinations:

A: location Vs gender B: Gender Vs frame C: Gender Vs Age (Convert age to an discrete ordinal variable with three categories) D: Cholesterol Vs Age (Convert age and cholesterol to an discrete ordinal variable with three categories)

calculate the joint and marginal probabilities, and from the above contingency tables choose 5 conditional probability examples with the probabilities calculations and one or two sentences explaining the results.

In [2]:
```
library("ggplot2")
library("dplyr")
library("reshape2")
library("knitr")
```

```
Attaching package: ‘dplyr’

The following objects are masked from ‘package:stats’:

    filter, lag

The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union
```

In [3]:
```
diabetes = read.csv(file = "diabetes.csv")
```

In [4]:
```
diabetes.location.gender.df <-
  diabetes %>%
  group_by(location, gender) %>%
  summarize(n = n())

diabetes.location.gender.prop.df <-
  diabetes.location.gender.df %>%
  ungroup() %>%
  mutate(prop = n / sum(n))

location.marginal.df <-
  diabetes.location.gender.prop.df %>%
  group_by(location) %>%
  summarize(marginal = sum(prop))

gender.marginal.df <-
  diabetes.location.gender.prop.df %>%
  group_by(gender) %>%
  summarize(marginal = sum(prop))


diabetes.location.gender.prop.df %>%
  dcast(location ~ gender, value.nar = "prop") %>%
  left_join(location.marginal.df, by = "location") %>%
  bind_rows(
    gender.marginal.df %>%
      mutate(location = "marginal") %>%
      dcast(location ~ gender, value.var = "marginal")
  ) %>%
  kable(align = "l", format = "markdown",
        table.attr='class="table table-striped table-hover"')
```

```
Using prop as value column: use value.var to override.
Warning message in bind_rows_(x, .id):
“binding factor and character vector, coercing into character vector”Warning message in
bind_rows_(x, .id):
“binding character and factor vector, coercing into character vector”
```

location	female	male	marginal
Buckingham	0.2828784	0.2133995	0.4962779
Louisa	0.2977667	0.2059553	0.5037221
marginal	0.5806452	0.4193548	NA

In [5]:
```
diabetes.gender.frame.df <-
  diabetes %>%
  group_by(gender, frame) %>%
  summarize(n = n())

diabetes.gender.frame.prop.df <-
  diabetes.gender.frame.df %>%
  ungroup() %>%
  mutate(prop = n / sum(n))

gender.marginal.df <-
  diabetes.gender.frame.prop.df %>%
  group_by(gender) %>%
  summarize(marginal = sum(prop))

frame.marginal.df <-
  diabetes.gender.frame.prop.df %>%
  group_by(frame) %>%
  summarize(marginal = sum(prop))


diabetes.gender.frame.prop.df %>%
  dcast(gender ~ frame, value.nar = "prop") %>%
  left_join(gender.marginal.df, by = "gender") %>%
  bind_rows(
    frame.marginal.df %>%
      mutate(gender = "marginal") %>%
      dcast(gender ~ frame, value.var = "marginal")
  ) %>%
  kable(align = "l", format = "markdown",
        table.attr='class="table table-striped table-hover"')
```

```
Using prop as value column: use value.var to override.
Warning message in bind_rows_(x, .id):
“binding factor and character vector, coercing into character vector”Warning message in
bind_rows_(x, .id):
“binding character and factor vector, coercing into character vector”
```

gender	Var.2	large	medium	small	marginal
female	0.0173697	0.1042184	0.2878412	0.1712159	0.5806452
male	0.0124069	0.1513648	0.1687345	0.0868486	0.4193548
marginal	0.0297767	0.2555831	0.4565757	0.2580645	NA

In [6]:
```
min(diabetes$age)
max(diabetes$age)
```

```
19
```

```
92
```

In [7]:
```
diabetes$agecat1<-cut(diabetes$age, c(0,30,60,100))
```

In [8]:
```
diabetes.gender.agecat1.df <-
  diabetes %>%
  group_by(gender, agecat1) %>%
  summarize(n = n())

diabetes.gender.agecat1.prop.df <-
  diabetes.gender.agecat1.df %>%
  ungroup() %>%
  mutate(prop = n / sum(n))

gender.marginal.df <-
  diabetes.gender.agecat1.prop.df %>%
  group_by(gender) %>%
  summarize(marginal = sum(prop))

agecat1.marginal.df <-
  diabetes.gender.agecat1.prop.df %>%
  group_by(agecat1) %>%
  summarize(marginal = sum(prop))


diabetes.gender.agecat1.prop.df %>%
  dcast(gender ~ agecat1, value.nar = "prop") %>%
  left_join(gender.marginal.df, by = "gender") %>%
  bind_rows(
    agecat1.marginal.df %>%
      mutate(gender = "marginal") %>%
      dcast(gender ~ agecat1, value.var = "marginal")
  ) %>%
  kable(align = "l", format = "markdown",
        table.attr='class="table table-striped table-hover"')
```

```
Using prop as value column: use value.var to override.
Warning message in bind_rows_(x, .id):
“binding factor and character vector, coercing into character vector”Warning message in
bind_rows_(x, .id):
“binding character and factor vector, coercing into character vector”
```

gender	(0,30]	(30,60]	(60,100]	marginal
female	0.1215881	0.3399504	0.1191067	0.5806452
male	0.0645161	0.2456576	0.1091811	0.4193548
marginal	0.1861042	0.5856079	0.2282878	NA

In [9]:
```
min(diabetes[complete.cases(diabetes), ]$chol)
max(diabetes[complete.cases(diabetes), ]$chol)
```

```
134
```

```
443
```

In [14]:
```
diabetes$cholcat<-cut(diabetes$chol, c(0,200,250,450))
```

In [17]:
```
diabetes.cholcat.agecat1.df <-
  diabetes %>%
  filter(cholcat != "NA") %>%
  group_by(cholcat, agecat1) %>%
  summarize(n = n())

diabetes.cholcat.agecat1.prop.df <-
  diabetes.cholcat.agecat1.df %>%
  ungroup() %>%
  mutate(prop = n / sum(n))

cholcat.marginal.df <-
  diabetes.cholcat.agecat1.prop.df %>%
  group_by(cholcat) %>%
  summarize(marginal = sum(prop))

agecat1.marginal.df <-
  diabetes.cholcat.agecat1.prop.df %>%
  group_by(agecat1) %>%
  summarize(marginal = sum(prop))


diabetes.cholcat.agecat1.prop.df %>%
  dcast(cholcat ~ agecat1, value.nar = "prop") %>%
  left_join(cholcat.marginal.df, by = "cholcat") %>%
  bind_rows(
    agecat1.marginal.df %>%
      mutate(cholcat = "marginal") %>%
      dcast(cholcat ~ agecat1, value.var = "marginal")
  ) %>%
  kable(align = "l", format = "markdown",
        table.attr='class="table table-striped table-hover"')
```

```
Using prop as value column: use value.var to override.
Warning message in bind_rows_(x, .id):
“binding factor and character vector, coercing into character vector”Warning message in
bind_rows_(x, .id):
“binding character and factor vector, coercing into character vector”
```

cholcat	(0,30]	(30,60]	(60,100]	marginal
(0,200]	0.1268657	0.2611940	0.0721393	0.4601990
(200,250]	0.0547264	0.2338308	0.1069652	0.3955224
(250,450]	0.0049751	0.0895522	0.0497512	0.1442786
marginal	0.1865672	0.5845771	0.2288557	NA

In [18]:
```
5 conditional probability examples with the probabilities calculations and one or two se
ntences explaining the results.

1. A: Cholesterol in (250, 450]
   B: Age in (0,30]
   P(A|B) = 0.0049751/0.1865672=0.027
   It shows only 2% younger people (0-30) have high level Cholesterol.

2. A: Cholesterol in (250, 450]
   B: Age in (30,100]
   P(A|B) = (0.0895522+0.0497512)/(0.5845771+0.2288557)=0.17
   It shows 17% people (30-100) have high level Cholesterol. From above probailites, we
 see that
   older people have a higher probability to get a high level cholesterol.

3.A: people live in Buckingham
B: female
P(A|B) = 0.2828784/0.5806452=0.48
There are 48% female who live in Buckingham

4.A: people live in Louisa
B: male
P(A|B) = 0.2059553/0.4193548=0.49
There are 49% male who live in Louisa

5. A: male
B: frame is large
P(A|B) = 0.1513648/0.2555831=0.59
There are 59% male in the people who have large frames
```

```
Error in parse(text = x, srcfile = src): <text>:1:3: unexpected symbol
1: 5 conditional
      ^
Traceback:
```

3. Baye's Rule

Write a function in R that allows you to use the Baye's rule for multiple events.

Use it to calculate the following problem:

Different isoforms of fundamental Hormones such as testosterone are relatively associated to behavioral differences in humans such as extreme aggression. In a study where 75 % of the participants had the Isoform A, and 25 % had the isoform B. 54 people exhibited extreme aggression out 95 that had Isoform A, and 34 people with extreme aggression out of 90 that had the isoform B.

Calculate the probability that someone with the isoform A exhibits extreme aggression. Calculate the probability that someone with the isoform B exhibits extreme aggression.

In [1]:
```
# pa is P(A), pba is P(B|A), pba_c is P(B|A^c)
bayes = function(pa,pba,pba_c){
  result = pa*pba/(pba*pa+pba_c*(1-pa))
  return(result)
}

A: someone with the isoform A
A^C: someone with the isoform B
B: exhibits extreme aggression

the probability that someone with the isoform A exhibits extreme aggression
p(A|B) = p(A)*P(B|A)/(P(B|A)P(A)+P(B|A^C)*P(A^C))=0.75*(54/95)/(54/95*0.75+34/90*0.25)=
0.82

the probability that someone with the isoform B exhibits extreme aggression.
p=0.25*(34/90)/(34/90*0.25+54/95*0.75)=0.18
```

```
Error in parse(text = x, srcfile = src): <text>:7:5: unexpected symbol
6:
7: the probability
       ^
Traceback:
```