Stat 5810, Section 003
Statistical Visualization I
Fall 2018

Dr. Jürgen Symanzik

Utah State University

Department of Mathematics and Statistics

3900 Old Main Hill

Logan, UT 84322–3900

Tel.: (435) 797–0696

FAX: (435) 797–1822

e-mail: symanzik@math.usu.edu

Web: http://www.math.usu.edu/~symanzik/

Contents

Acknowledgements

This course uses some of the course materials provided by Dr. Mike Minnotte (formerly USU, now with the University of North Dakota) as held in the Fall 2006 semester. Additional materials have been taken from other Statistical Graphics courses, such as the ones offered by Dr. Di Cook (formerly Iowa State University; now Monash University: `http://dicook.org/`) and Dr. Dan Carr (George Mason University: `http://mason.gmu.edu/~dcarr/`). Other examples and R code originate from Heike Hofmann, Paul Murrell, Carson Sievert, Martin Theus, Antony Unwin, Simon Urbanek, Hadley Wickham, Lee Wilkinson, and others. We are likely to include parts from additional authors and sources that will be specified later during the semester.

Thanks are also due to 60+ students and guests who took the former "Stat 6560: Graphical Methods" and the current "Statistical Visualization I & II" courses with me since the Spring 2009 semester for their valuable comments that helped to improve, correct, and extend these lecture notes.

Jürgen Symanzik, September 10, 2018.

1 Categorical Plots

1.1 Which Plot Type to Choose?

Often, there exist many valid options how to display (categorical) data.

?, p. 12, suggests the following project:

> "Sketch as many charts as you can think of using these data: the more the better."

Percentage of January Sales by Region

	Co. A	Co. B
North	13%	39%
South	35%	6%
East	27%	27%
West	25%	28%

Worksheet

Your Name: _____

Worksheet Answers

1.2 Categorical Plots in R

Recall Section 2.4, *"Sex Bias in Graduate Admissions"*, from **?**, pp. 17–20, many of us are using in our introductory Stat 1040 class.

These data represent aggregate data on applicants to graduate school at Berkeley for the six largest departments in 1973, classified by admission and sex. These data are often used to discuss the issue whether the data show evidence of sex bias in admission practices. There were 2691 male applicants, of whom 1198 (44.5%) were admitted, compared with 1835 female applicants of whom 557 (30.4%) were admitted. Ultimately, this data set is frequently used for illustrating Simpson's paradox and does not show any sex bias when properly analyzed. An effective graphical way to explain Simpson's Paradox is the BK–Plot, summarized in **?**.

In R, the data are stored in a a 3–dimensional array resulting from cross–tabulating 4526 observations on 3 variables. The variables and their levels are as follows:

No	Name	Levels
1	Admit	Admitted, Rejected
2	Gender	Male, Female
3	Dept	A, B, C, D, E, F

In R, this data set is accessible via:

```
> UCBAdmissions

, , Dept = A

          Gender
Admit      Male Female
  Admitted  512     89
  Rejected  313     19

, , Dept = B

          Gender
```

```
Admit      Male Female
  Admitted  353     17
  Rejected  207      8

, , Dept = C

          Gender
Admit      Male Female
  Admitted  120    202
  Rejected  205    391

, , Dept = D

          Gender
Admit      Male Female
  Admitted  138    131
  Rejected  279    244

, , Dept = E

          Gender
Admit      Male Female
  Admitted   53     94
  Rejected  138    299

, , Dept = F

          Gender
Admit      Male Female
  Admitted   22     24
  Rejected  351    317
```

A better tabular representation can be obtained via:

```
> ftable(UCBAdmissions)

                Dept   A   B   C   D   E   F
Admit    Gender
```

```
Admitted Male        512 353 120 138  53  22
         Female       89  17 202 131  94  24
Rejected Male        313 207 205 279 138 351
         Female       19   8 391 244 299 317
```

To obtain the totals as represented in ?, p. 18, we have to sum over dimensions 2 and 3 in this 3–dimensional array:

```
> apply(UCBAdmissions, c(2, 3), sum)

        Dept
Gender     A   B   C   D   E   F
  Male   825 560 325 417 191 373
  Female 108  25 593 375 393 341

> #
> # also, margin.table prodoces the same result
> #
> margin.table(UCBAdmissions, 2:3)

        Dept
Gender     A   B   C   D   E   F
  Male   825 560 325 417 191 373
  Female 108  25 593 375 393 341
```

To better understand over which dimensions we sum, replace the `c(2, 3)` option with possible other indices, e.g., `1` or `c(1, 2)`. Try a few more.

Question:

How can we calculate in R the percent admitted, as shown in **?**, p. 18, Table 2? This can be done via a single command line and does not require any loop! And, which single digit do we have to change in our previous R command to obtain the percent rejected?

Answer:

```
> # Percent admitted
> UCBAdmissions[1, , ] / apply(UCBAdmissions, c(2, 3), sum) * 100

        Dept
Gender           A         B         C         D         E         F
  Male   62.060606 63.035714 36.923077 33.093525 27.748691  5.898123
  Female 82.407407 68.000000 34.064081 34.933333 23.918575  7.038123

> # Percent rejected
> UCBAdmissions[2, , ] / apply(UCBAdmissions, c(2, 3), sum) * 100

        Dept
Gender           A        B        C        D        E        F
  Male   37.93939 36.96429 63.07692 66.90647 72.25131 94.10188
  Female 17.59259 32.00000 65.93592 65.06667 76.08142 92.96188
```

1.2.1 Pie Charts

Let us concentrate on the popularity of the six majors first, i.e., the total number of admissions for each of these majors.

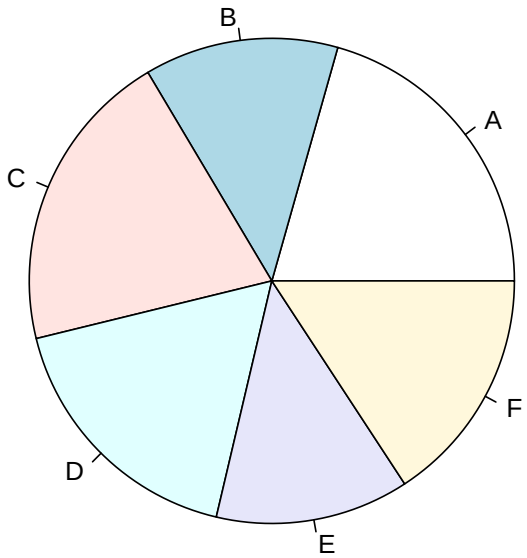
In R, these application numbers can be calculated via:

```
> apply(UCBAdmissions, 3, sum)
```

```
  A   B   C   D   E   F
933 585 918 792 584 714
```

Many people would immediately think of a pie chart as a possible graphical representation:

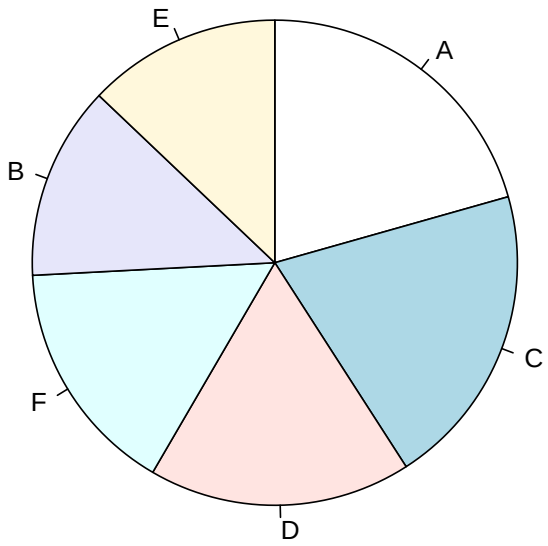
```
> pie(apply(UCBAdmissions, 3, sum))
```



Note that there is no sorting here. Can you easily order the slices by visual inspection, i.e., which major has the largest number/percentage of admissions, which is second, third, etc.?

A better representation is to sort the data from largest to smallest and then plot the slices in clockwise direction, starting with the largest slice at 90°.

```
> pie(sort(apply(UCBAdmissions, 3, sum), decreasing = TRUE),
+     clockwise = TRUE,
+     main = "UC Berkley Admissions by Major")
```



This is somewhat better, but still not perfect. The R help page for pie charts indicates:

> "Pie charts are a very bad way of displaying information. The eye is good at judging linear measures and bad at judging relative areas. A bar chart or dot chart is a preferable way of displaying this type of data."

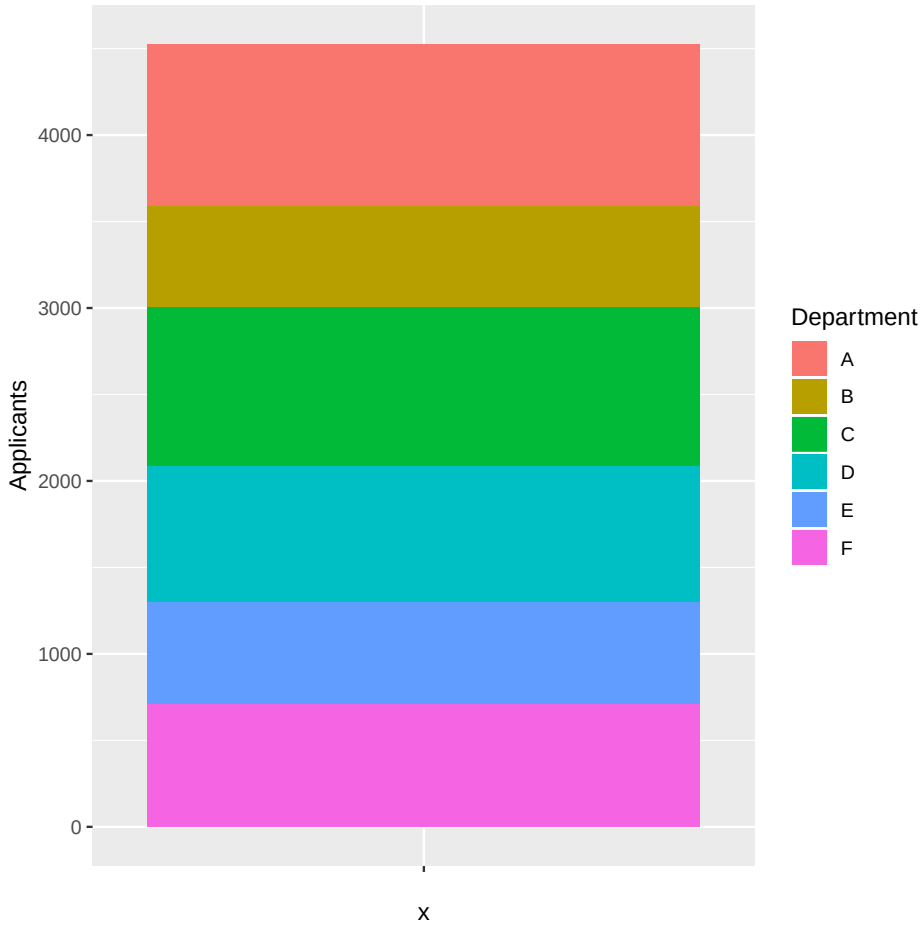
Moreover, **?**, p. 264, states:

> "Data that can be shown by pie charts always can be shown by a dot chart. This means that judgements of position along a common scale can be made instead of the less accurate angle judgements."

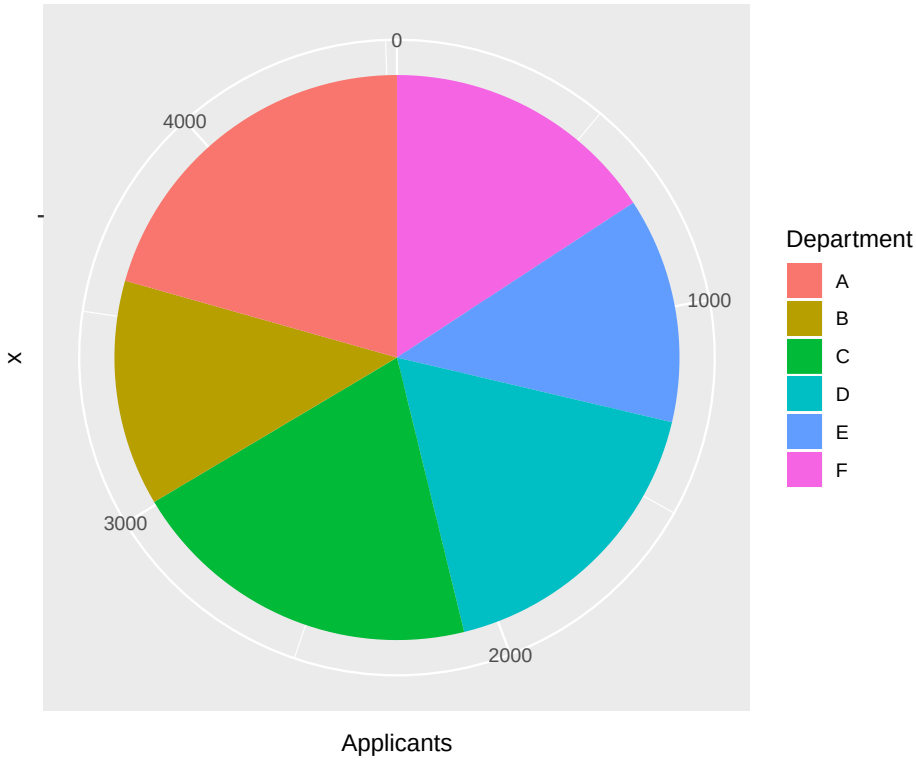
ggplot2 allows to draw pie charts, but it is not very supportive. The help page for `coord_polar` states:

> NOTE: Use these plots with caution - polar coordinates has major perceptual problems. The main point of these examples is to demonstrate how these common plots can be described in the grammar. Use with EXTREME caution.

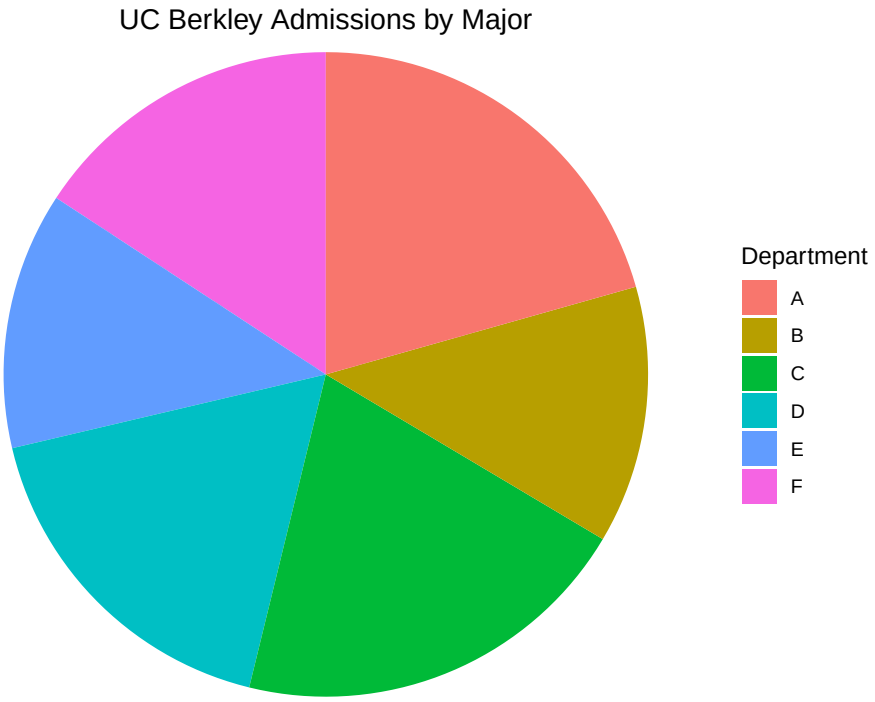
```
> library(ggplot2)
> UCBtable <- apply(UCBAdmissions, 3, sum)
> UCBdf <- data.frame(Department = as.factor(names(UCBtable)),
+                     Applicants = UCBtable)
> UCBdf

  Department Applicants
A          A        933
B          B        585
C          C        918
D          D        792
E          E        584
F          F        714

> # stacked bar chart
> bp <- ggplot(UCBdf, aes(x = "", y = Applicants, fill = Department)) +
+   geom_bar(width = 1, stat = "identity")
> bp
```




```
> # pie chart
> pie <- bp + coord_polar(theta = "y")
> pie
```




```
> # modified pie chart
> pie <- ggplot(UCBdf, aes(x = "", y = Applicants, fill = Department)) +
+   geom_bar(width = 1, stat = "identity") +
+   coord_polar(theta = "y", start = 0, direction = -1) +
+   theme_void() +
+   ggtitle("UC Berkley Admissions by Major") +
+   theme(plot.title = element_text(hjust = 0.5, vjust = -10))
> pie
```



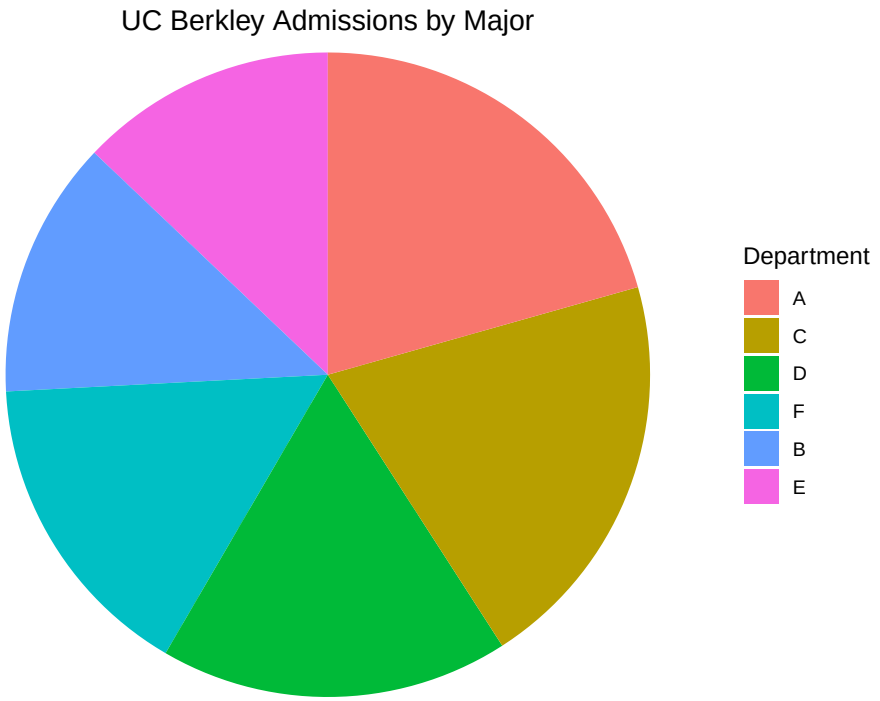

```
> # modified pie chart, sorted from largest to smallest segment
> UCBdfsort <- UCBdf
> UCBdfsort$Department

[1] A B C D E F
Levels: A B C D E F

> UCBdfsort$Department <- factor(UCBdfsort$Department,
+                                levels = UCBdfsort$Department[order(-UCBdfsort$Applic
> UCBdfsort

  Department Applicants
A          A        933
B          B        585
C          C        918
```

```
D          D        792
E          E        584
F          F        714
```

```
> UCBdfsort$Department
```

```
[1] A B C D E F
Levels: A C D F B E
```

```
> pie <- ggplot(UCBdfsort, aes(x = "", y = Applicants, fill = Department)) +
+   geom_bar(width = 1, stat = "identity") +
+   coord_polar(theta = "y", start = 0, direction = -1) +
+   theme_void() +
+   ggtitle("UC Berkley Admissions by Major") +
+   theme(plot.title = element_text(hjust = 0.5, vjust = -10))
> pie
```



If you further want to adjust your pie chart via *ggplot2*, see the `http://www.dxbydt.com/bars-pies-using-ggplot2/` web page for additional suggestions.

And what about the extremely popular 3D–pie charts that often can be found in business reports and the media? The answer is a clear **Don't.**

?, p. 70, provide a striking example why not to use 3D–pie charts. Guess the percentages associated with the four different areas:

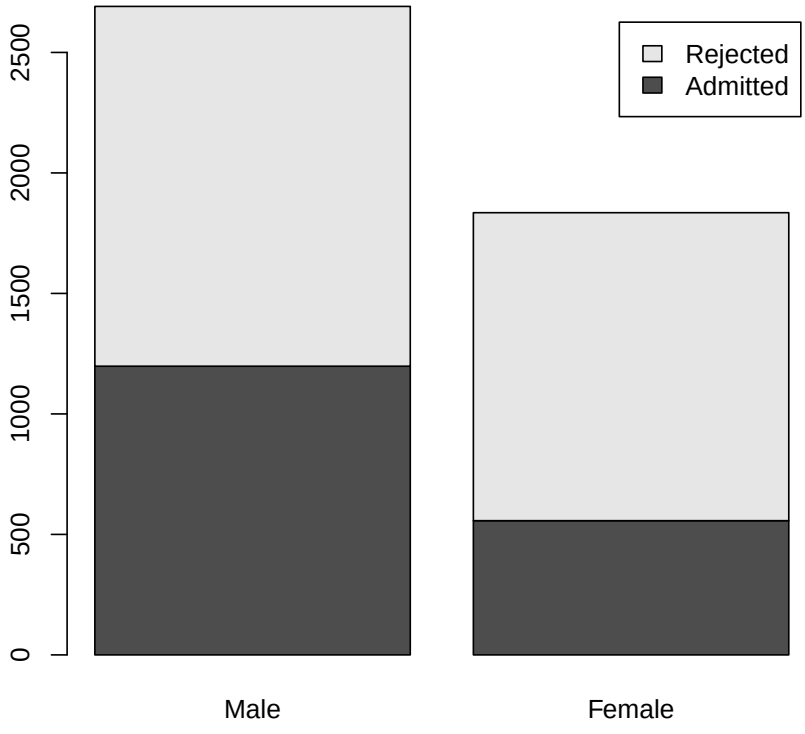
And here is the answer:

1.2.2 Bar Charts

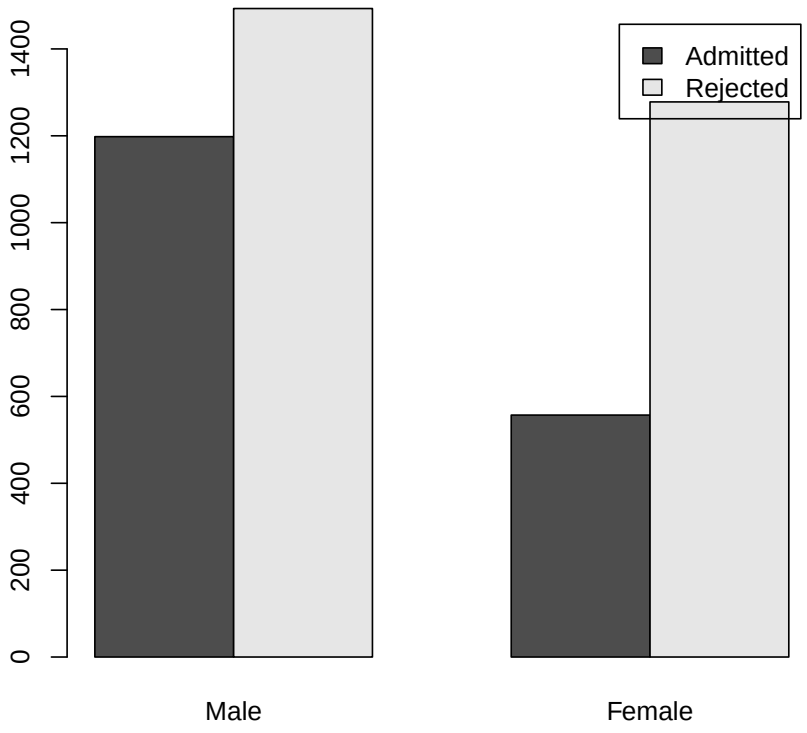
The R help page for barplot indicates:

> "Creates a bar plot with vertical or horizontal bars."

```
> UCBAd <- margin.table(UCBAdmissions, 1:2)
> UCBAd

          Gender
Admit      Male Female
  Admitted 1198    557
  Rejected 1493   1278

> barplot(UCBAd, legend.text = TRUE)
```

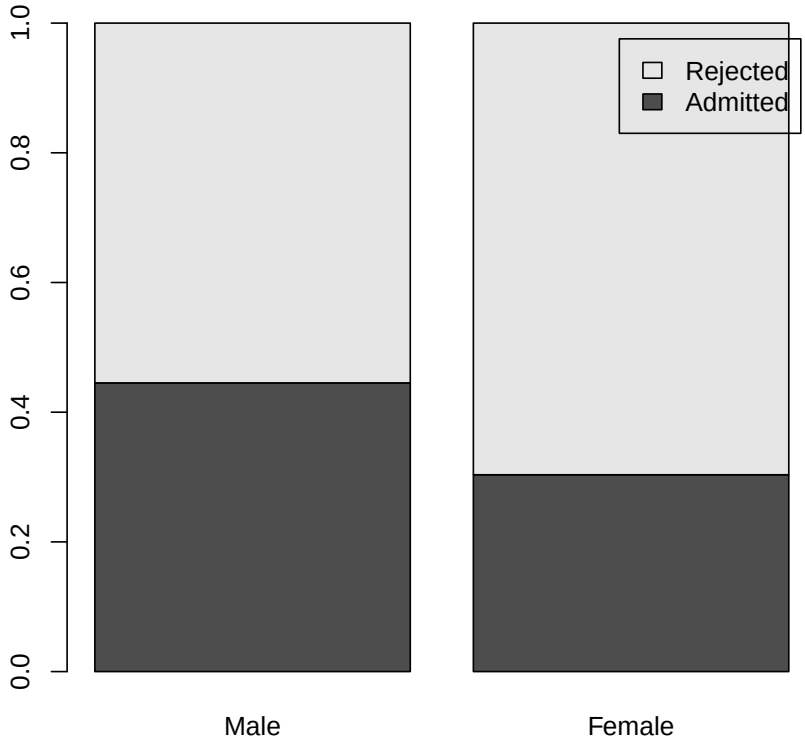


```
> barplot(UCBAd, legend.text = TRUE, beside = TRUE)
```

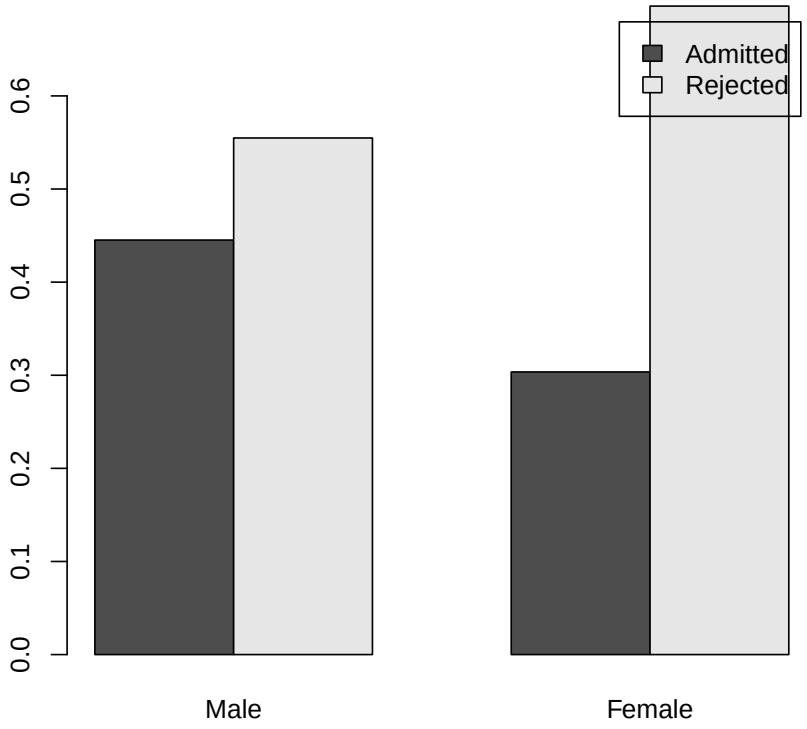


The following commands create (divided) bar charts that show the percentage admitted/rejected for each gender.

```
> barplot(UCBAd / rbind(margin.table(UCBAd, 2), margin.table(UCBAd, 2)),
+   legend.text = TRUE)
```




```
> barplot(UCBAd / rbind(margin.table(UCBAd, 2), margin.table(UCBAd, 2)),
+   legend.text = TRUE, beside = TRUE)
```



Let's try a few more bar charts with ggplot2. Thanks to Johnny Hong for posting these examples at `http://jcyhong.github.io/ggplot_demo.html`:

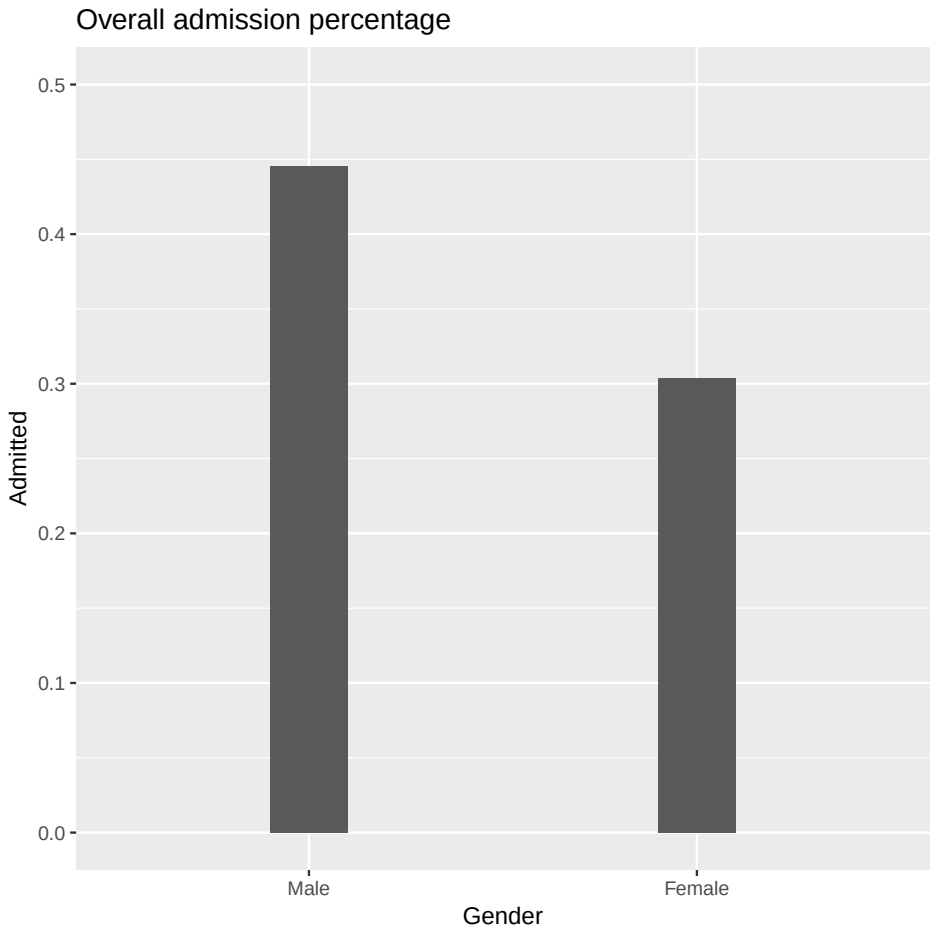
```
> library(ggplot2)
> library(plyr)
> library(gridExtra)
> library(grid)
> UCBdt <- as.data.frame(UCBAdmissions)
> overall <- ddply(UCBdt, .(Gender), function(gender) {
+   temp <- c(sum(gender[gender$Admit == "Admitted", "Freq"]),
+             sum(gender[gender$Admit == "Rejected", "Freq"])) / sum(gender$Freq)
+   names(temp) <- c("Admitted", "Rejected")
+   temp
+ })
> overall
```

```
  Gender  Admitted  Rejected
1   Male 0.4451877 0.5548123
2 Female 0.3035422 0.6964578
```

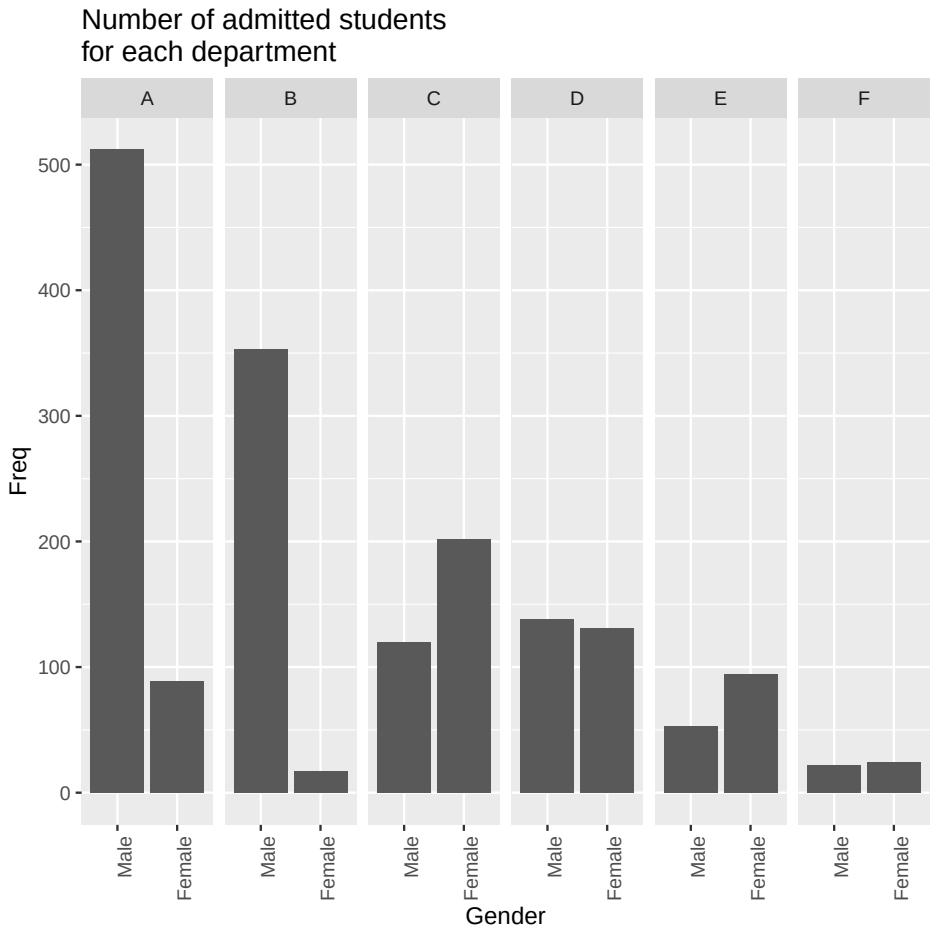
```
> departmentwise <- ddply(UCBdt, .(Gender, Dept), function(gender) {
+   temp <- gender$Freq / sum(gender$Freq)
+   names(temp) <- c("Admitted", "Rejected")
+   temp
+ })
> departmentwise
```

```
   Gender Dept   Admitted  Rejected
1    Male    A 0.62060606 0.3793939
2    Male    B 0.63035714 0.3696429
3    Male    C 0.36923077 0.6307692
4    Male    D 0.33093525 0.6690647
5    Male    E 0.27748691 0.7225131
6    Male    F 0.05898123 0.9410188
7  Female    A 0.82407407 0.1759259
8  Female    B 0.68000000 0.3200000
9  Female    C 0.34064081 0.6593592
10 Female    D 0.34933333 0.6506667
11 Female    E 0.23918575 0.7608142
12 Female    F 0.07038123 0.9296188
```

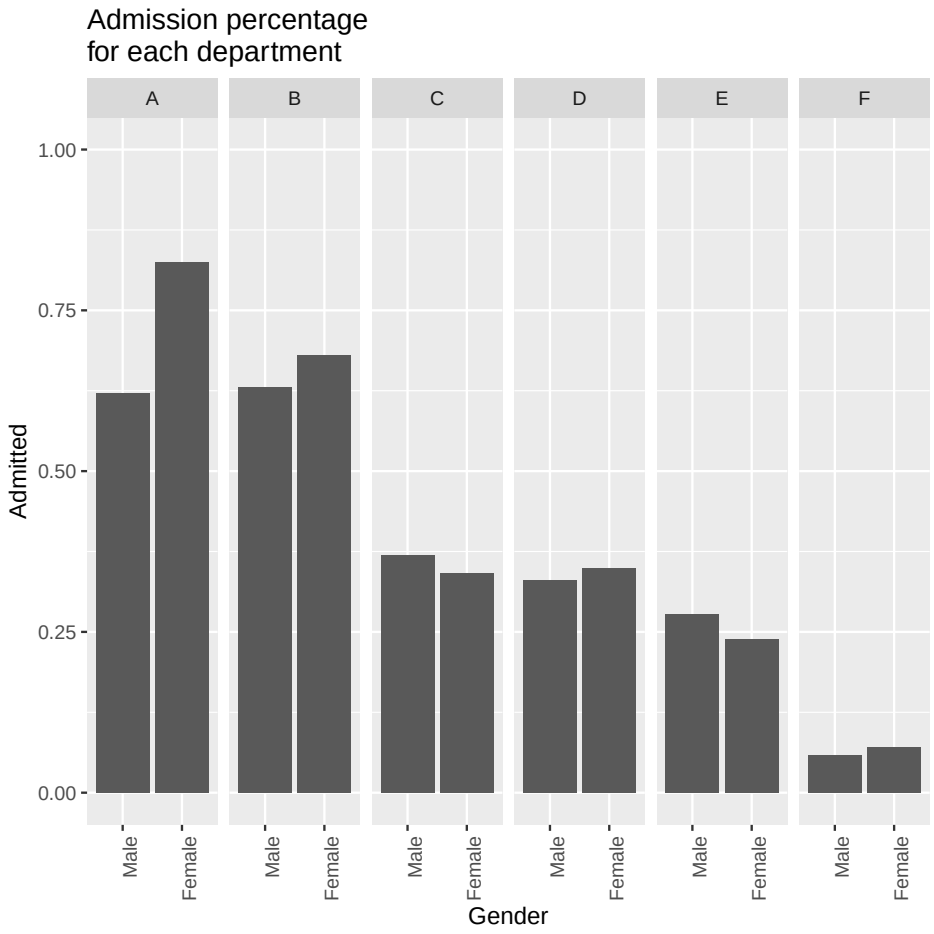
```
> # A barplot for overall admission percentage for each gender.
> p1 <- ggplot(data = overall, aes(x = Gender, y = Admitted, width = 0.2))
> p1 <- p1 + geom_bar(stat = "identity") +
+   ggtitle("Overall admission percentage") +
+   ylim(0, 0.5)
> p1
```



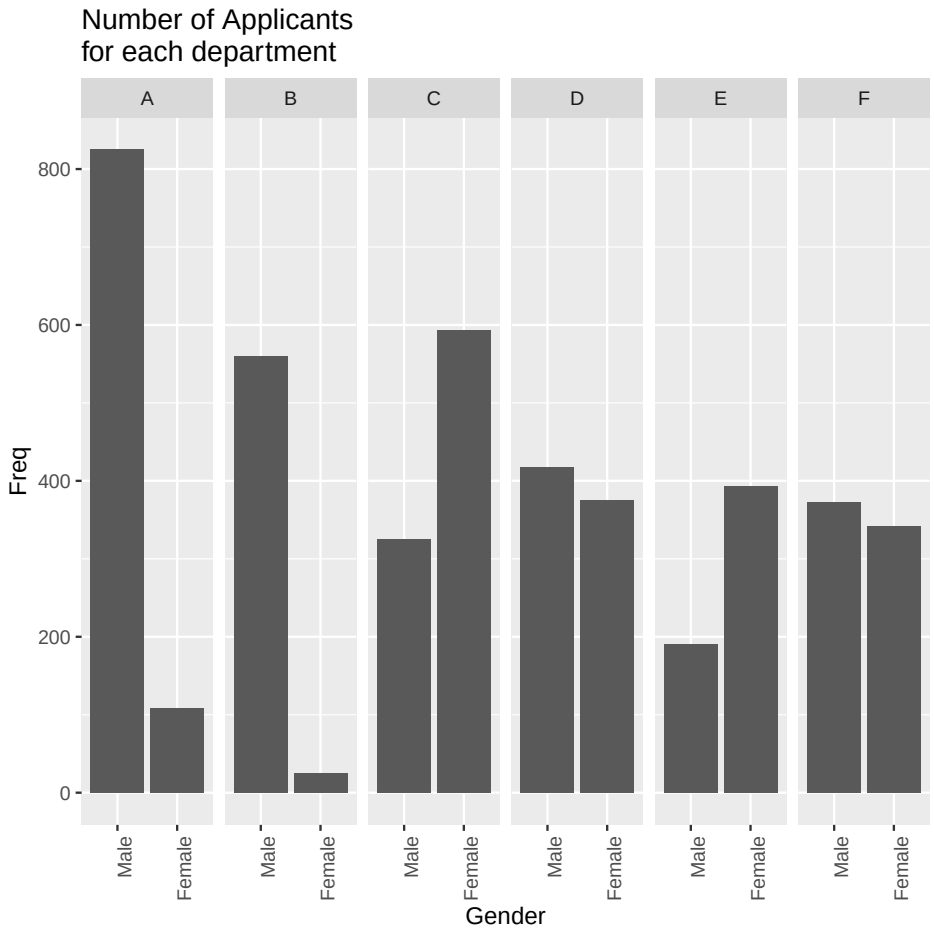

```
> # A 1x6 panel of barplots, each of which represents the
> # number of admitted students for a department
> p2 <- ggplot(data = UCBdt[UCBdt$Admit == "Admitted", ], aes(x = Gender, y = Freq))
> p2 <- p2 + geom_bar(stat = "identity") +
+   facet_grid(. ~ Dept) +
+   ggtitle("Number of admitted students\nfor each department") +
+   theme(axis.text.x = element_text(angle = 90, hjust = 1))
> p2
```



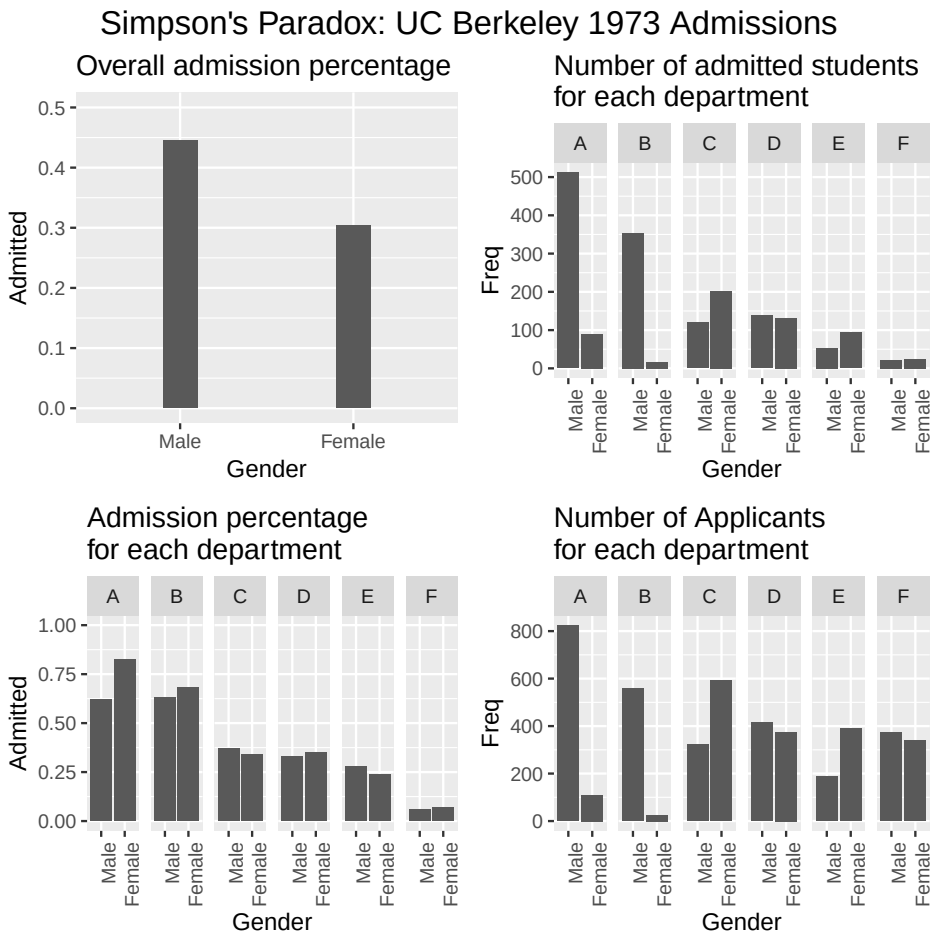

```
> # A 1x6 panel of barplots, each of which represents the
> # admission percentage for a department
> p3 <- ggplot(data = departmentwise, aes(x = Gender, y = Admitted))
> p3 <- p3 + geom_bar(stat = "identity") +
+   facet_grid(. ~ Dept) +
+   ylim(0, 1) +
+   ggtitle("Admission percentage\nfor each department") +
+   theme(axis.text.x = element_text(angle = 90, hjust = 1))
> p3
```




```
> # A 1x6 panel of barplots, each of which represents the
> # number of applicants for a department
> p4 <- ggplot(data = UCBdt, aes(x = Gender, y = Freq))
> p4 <- p4 + geom_bar(stat = "identity") +
+   facet_grid(. ~ Dept) +
+   ggtitle("Number of Applicants\nfor each department") +
+   theme(axis.text.x = element_text(angle = 90, hjust = 1))
> p4
```

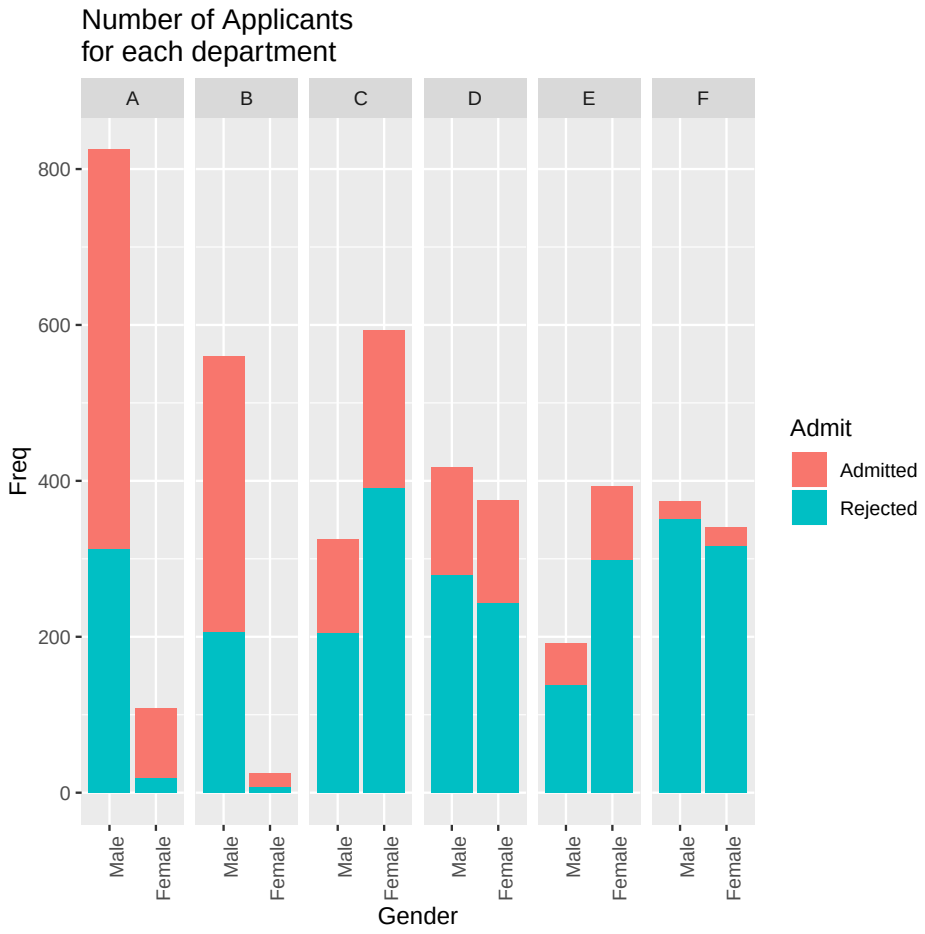



```
> # Arrange the four plots on a page
> grid.arrange(p1, p2, p3, p4, nrow = 2,
+              top = textGrob("Simpson's Paradox: UC Berkeley 1973 Admissions",
+                             gp = gpar(fontsize = 15)))
```

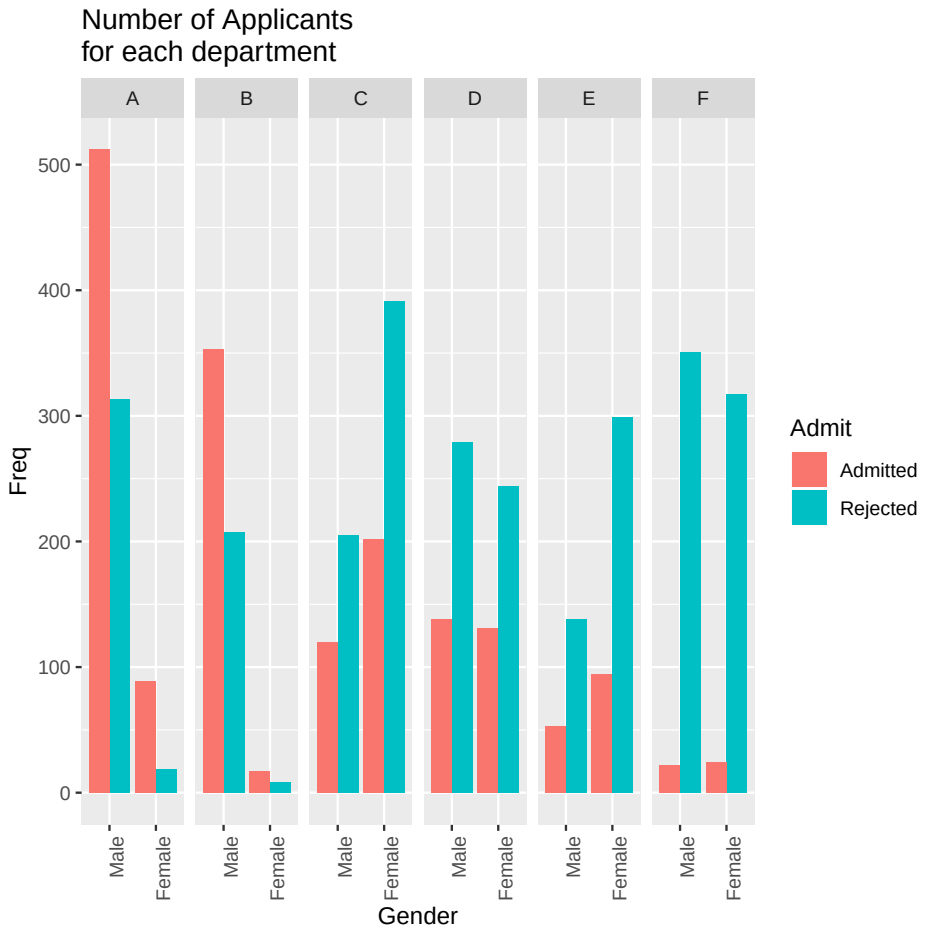


Blogger "wszafranski" provided further suggestions how to modify bar charts in ggplot2 at `https://www.r-bloggers.com/make-a-bar-plot-with-ggplot/`. Let's apply some of these to the previously created bar charts and compare them side-by-side.

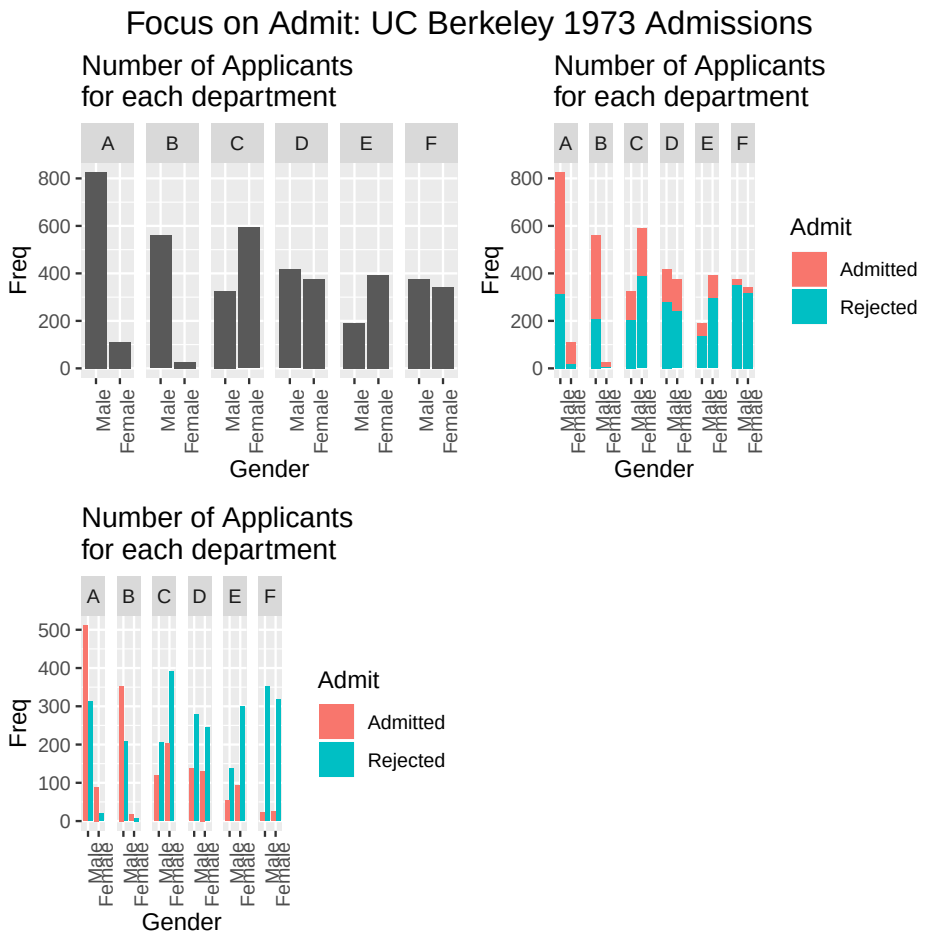
```
> # A 1x6 panel of barplots, each of which represents the
> # number of applicants for a department;
> # moreover, make a distinction between Admitted & Rejected
> p5 <- ggplot(data = UCBdt, aes(x = Gender, y = Freq))
> p5 <- p5 + geom_bar(stat = "identity", aes(fill = Admit)) +
+   facet_grid(. ~ Dept) +
+   ggtitle("Number of Applicants\nfor each department") +
+   theme(axis.text.x = element_text(angle = 90, hjust = 1))
> p5
```




```
> # A 1x6 panel of barplots, each of which represents the
> # number of applicants for a department;
> # moreover, make a distinction between Admitted & Rejected
> # and place these bars side-by-side
> p6 <- ggplot(data = UCBdt, aes(x = Gender, y = Freq))
> p6 <- p6 + geom_bar(stat = "identity", aes(fill = Admit), position = "dodge") +
+   facet_grid(. ~ Dept) +
+   ggtitle("Number of Applicants\nfor each department") +
+   theme(axis.text.x = element_text(angle = 90, hjust = 1))
> p6
```




```
> # Arrange the three plots on a page
> grid.arrange(p4, p5, p6, nrow = 2,
+              top = textGrob("Focus on Admit: UC Berkeley 1973 Admissions",
+                             gp = gpar(fontsize = 15)))
```



Warning:

?, Section 4.10, "Pop Charts", p. 262, strongly advises against the use of pie charts, divided bar charts, and area charts:

> Three graphical methods — pie charts, divided bar charts, and area charts — are widely used in mass media and business publications but are used far less in science and technology. Because of their use, we will call these graphical methods *pop charts*.
>
> Any data that can be encoded by one of these pop charts can also be encoded by either a dot plot or a multiway dot plot that typically provides far more efficient pattern perception and table look–up than the pop–chart encoding. Interestingly, the better pattern perception results from a detection operation, a phenomenon that has been missed in previous studies of pop charts.

Finally, what about 3D bar charts? **?** discussed problems with 3D bar charts in detail. They stated:

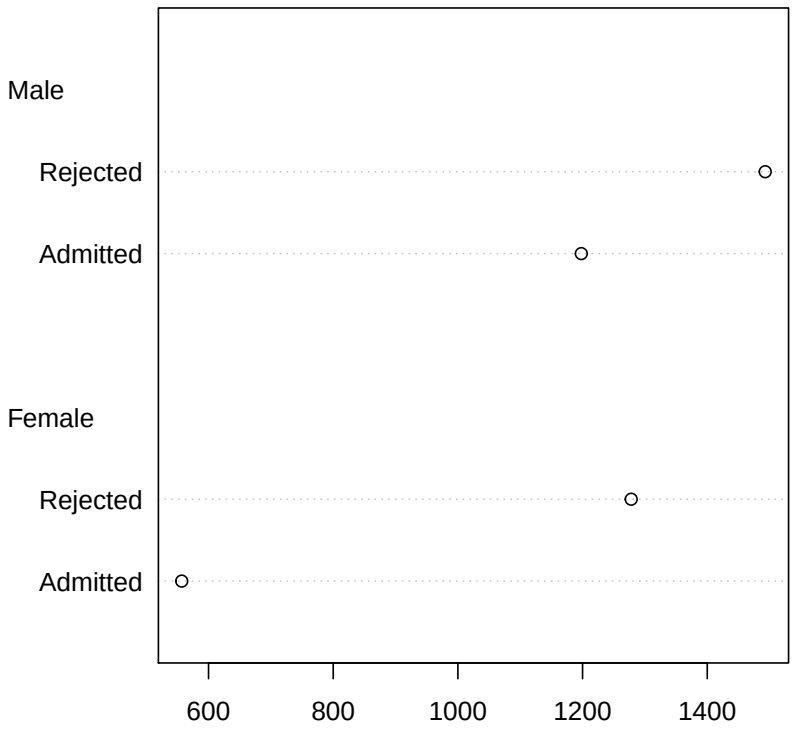
> "There are several reasons that we object to these 3D charts. Our main objection is that most people are misled by them. They don't know where the value is encoded. In Figure 11 (left), is the value at the front of the bar as indicated by the arrow on the A bar or is it in the back of the bar as indicated by the arrow on the B bar? Most readers judge the A bar to be around 0.75 and the B bar to be about 1.75. [...]
> Our second objection to pseudo 3D graphs is that different software programs draw them differently. PowerPoint is in the same suite of programs as Excel but earlier versions of PowerPoint did have a default gap depth of zero. Therefore, the usual value of a 3D bar graph drawn using these versions of PowerPoint was read from the back of the bar. Figure 12 (left) shows a figure from PowerPoint 2003 (not the same data as in Figures 11). A number of software programs show the value from the front of the bar as in the arrow of the A bar in Figure 11 and in Figure 12 (right). Even different versions of the same software may have different defaults. We find it unacceptable that the way to read a figure should depend on the software used to draw it. Readers often don't know what software was used and even if they did, they probably don't know the algorithm used."

1.2.3 Dot Plots / Dot Charts

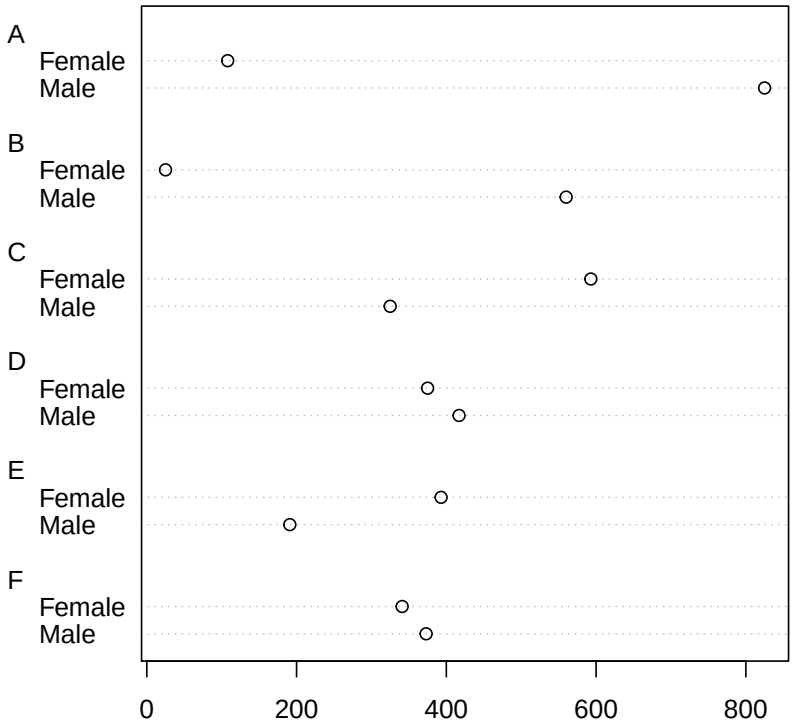
The R help page for dotchart indicates:

> "Draw a Cleveland dot plot. [...]
> This function is invoked for its side effect, which is to produce two variants of dotplots as described in **?**. Dot plots are a reasonable substitute for bar plots."

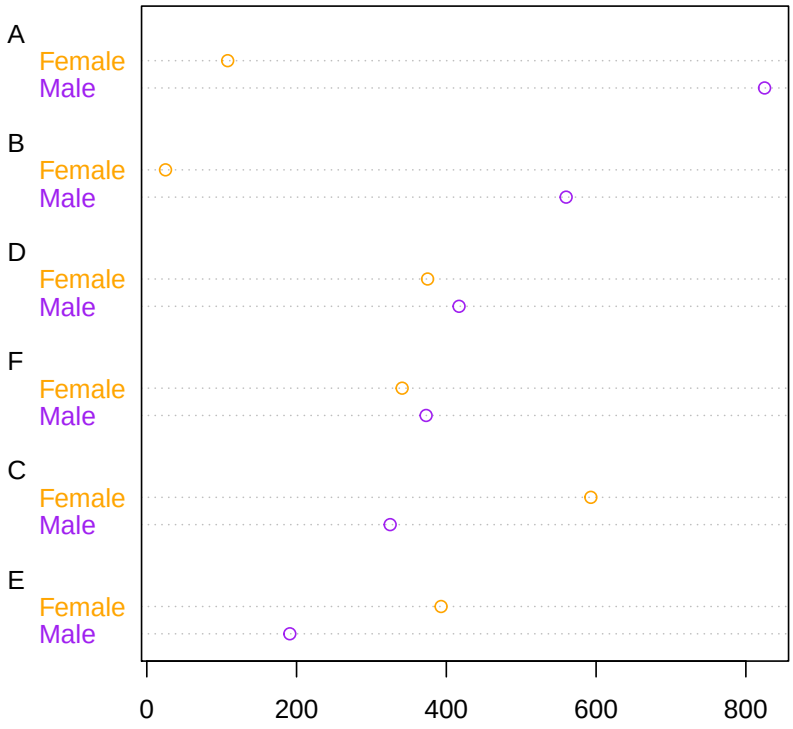
```
> dotchart(UCBAd)
```




```
> UCBMajor <- margin.table(UCBAdmissions, 2:3)
> dotchart(UCBMajor)
```



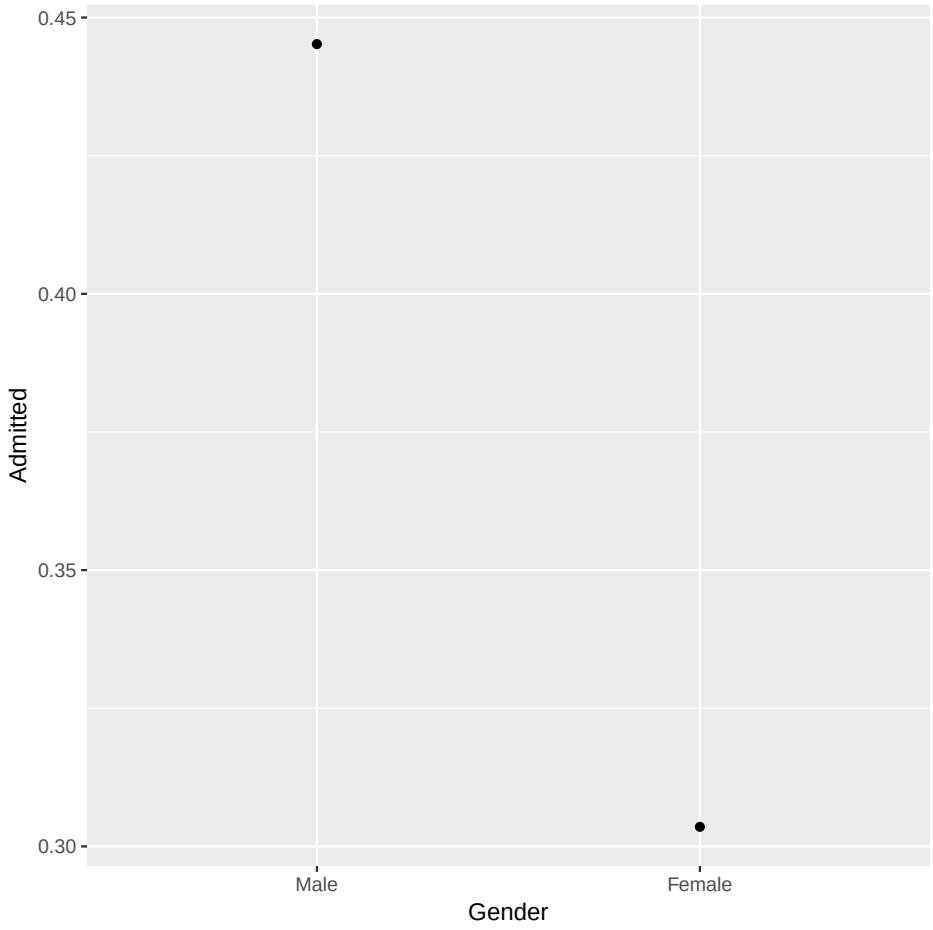

```
> UCBMajorsort <- UCBMajor[, order(UCBMajor[1, ], decreasing = TRUE)]
> dotchart(UCBMajorsort, color = c("purple", "orange"))
```



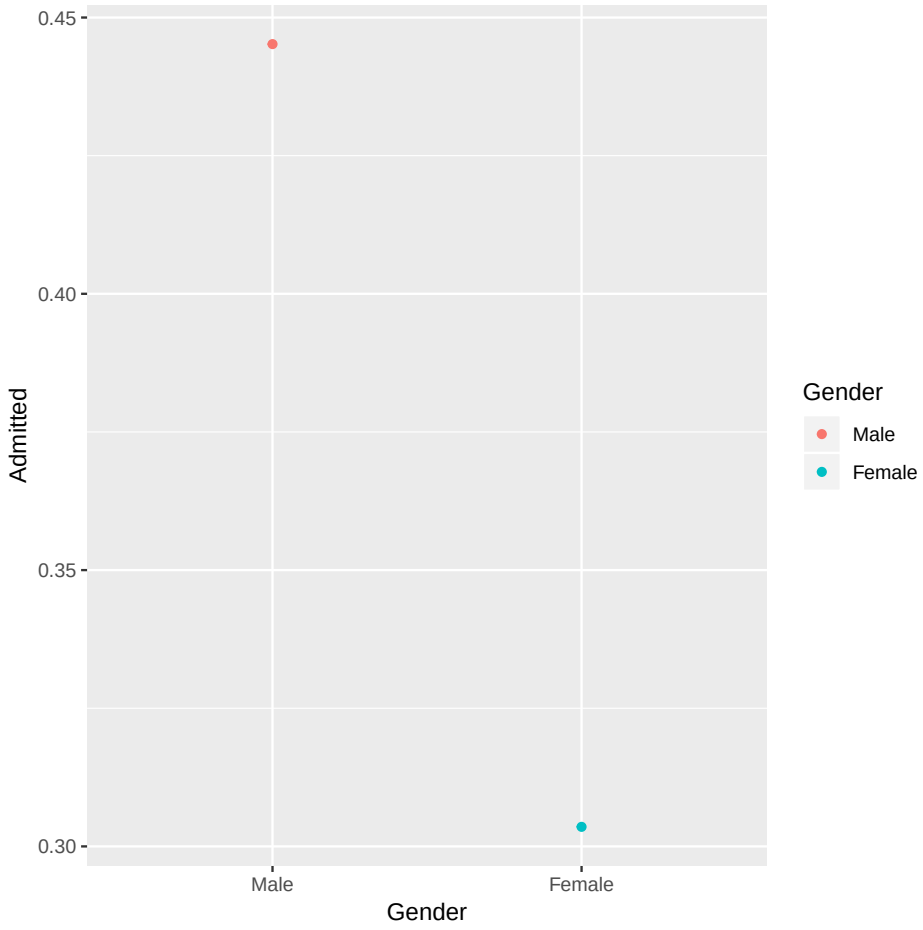
Let's try a few more dot plots with ggplot2.

```
> library(ggplot2)
> library(plyr)
> library(gridExtra)
> library(grid)
> UCBdt <- as.data.frame(UCBAdmissions)
> overall <- ddply(UCBdt, .(Gender), function(gender) {
+   temp <- c(sum(gender[gender$Admit == "Admitted", "Freq"]),
+             sum(gender[gender$Admit == "Rejected", "Freq"])) / sum(gender$Freq)
+   names(temp) <- c("Admitted", "Rejected")
+   temp
+ })
> overall

  Gender  Admitted  Rejected
```

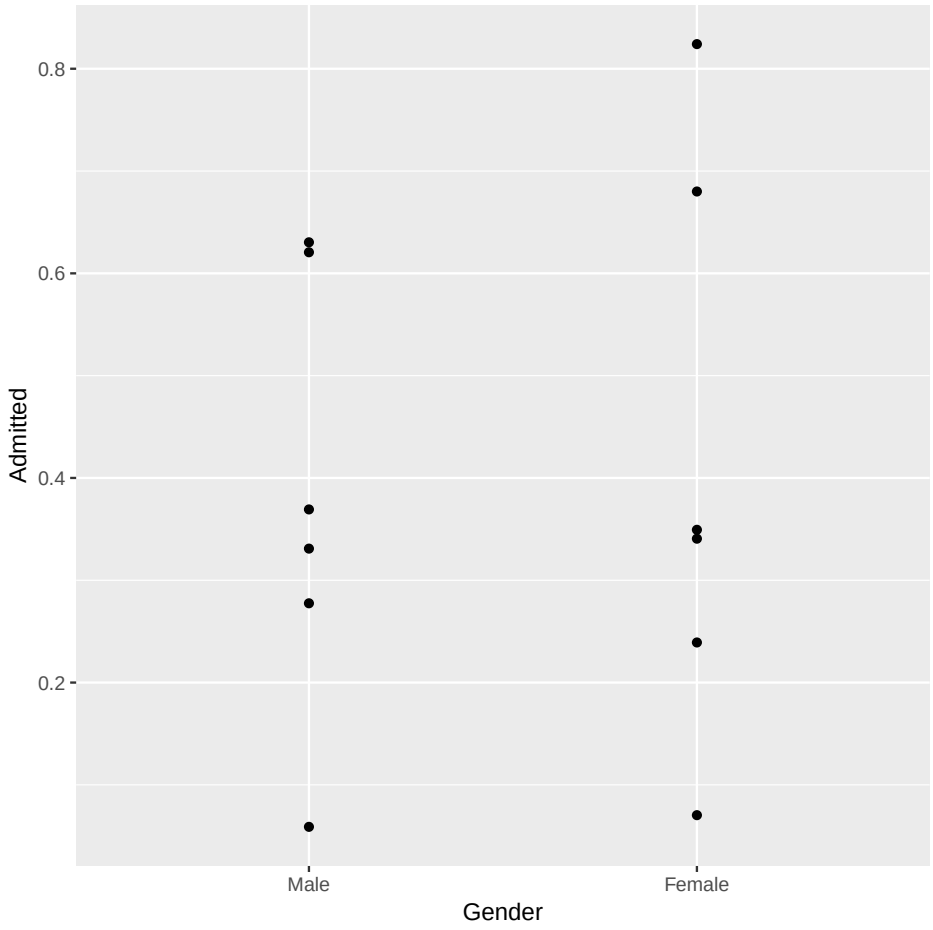
```
1   Male 0.4451877 0.5548123
2 Female 0.3035422 0.6964578

> departmentwise <- ddply(UCBdt, .(Gender, Dept), function(gender) {
+   temp <- gender$Freq / sum(gender$Freq)
+   names(temp) <- c("Admitted", "Rejected")
+   temp
+ })
> departmentwise

   Gender Dept   Admitted  Rejected
1    Male    A 0.62060606 0.3793939
2    Male    B 0.63035714 0.3696429
3    Male    C 0.36923077 0.6307692
4    Male    D 0.33093525 0.6690647
5    Male    E 0.27748691 0.7225131
6    Male    F 0.05898123 0.9410188
7  Female    A 0.82407407 0.1759259
8  Female    B 0.68000000 0.3200000
9  Female    C 0.34064081 0.6593592
10 Female    D 0.34933333 0.6506667
11 Female    E 0.23918575 0.7608142
12 Female    F 0.07038123 0.9296188

> # 3 alternative ways for the same graph
>
> ggplot(data = overall, aes(x = Gender, y = Admitted)) +
+   geom_point()
> # or
>
> ggplot(data = overall) +
+   geom_point(aes(x = Gender, y = Admitted))
> # or
>
> qplot(Gender, Admitted, data = overall)
>
> # Note: qplot is a shortcut that can be used for some graphs in ggplot
```



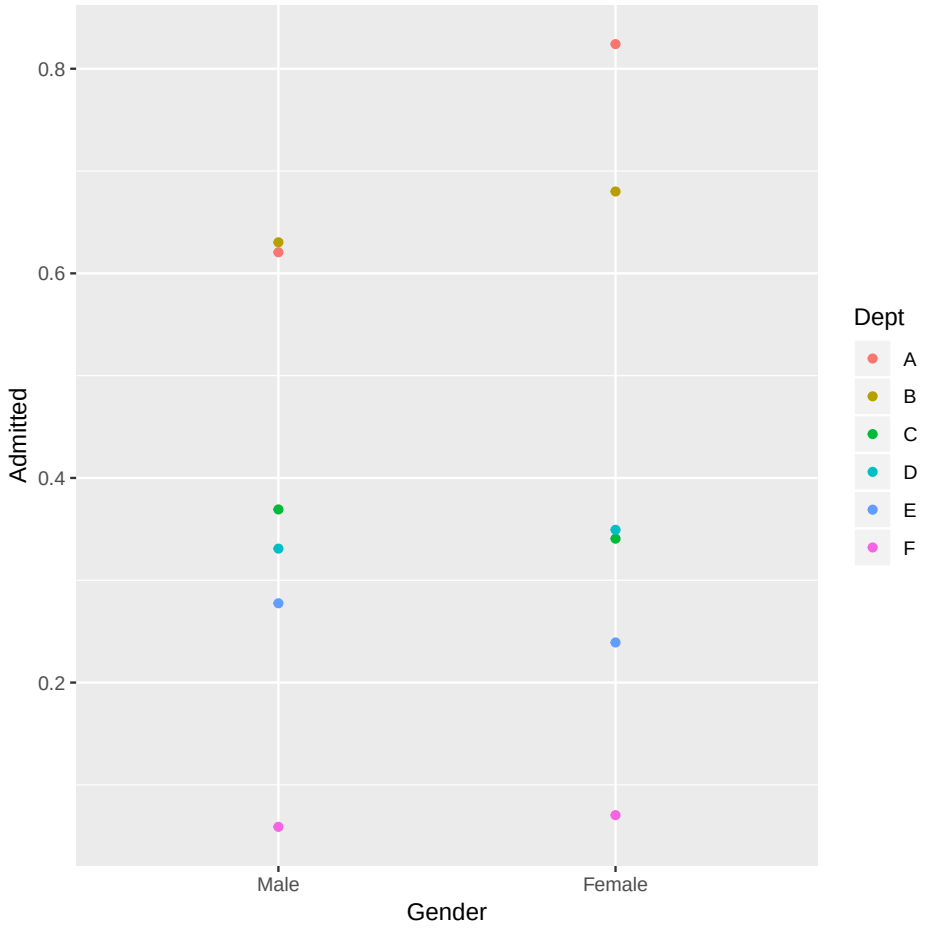

```
> ggplot(data = overall, aes(x = Gender, y = Admitted, colour = Gender)) +
+   geom_point()
> # or
>
> ggplot(data = overall) +
+   geom_point(aes(x = Gender, y = Admitted, colour = Gender))
> # or
>
> qplot(Gender, Admitted, data = overall, colour = Gender)
```



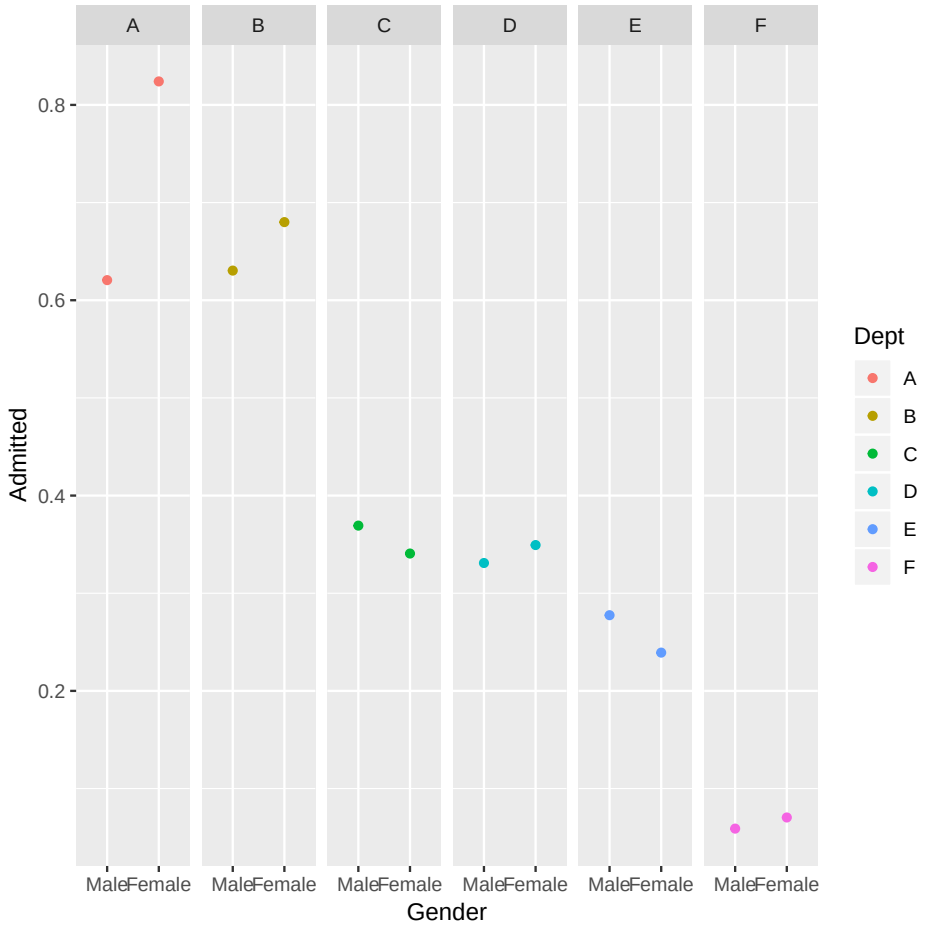

```
> ggplot(data = departmentwise) +
+   geom_point(aes(x = Gender, y = Admitted))
> # or
>
> qplot(Gender, Admitted, data = departmentwise)
```



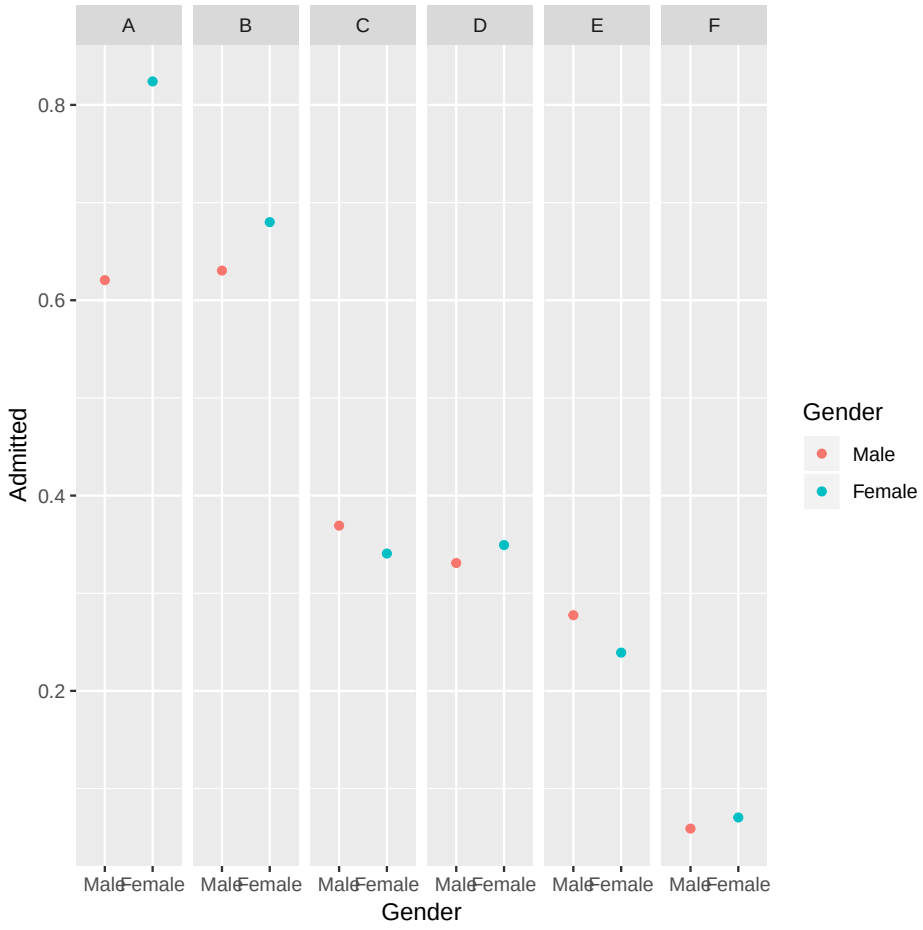

```
> ggplot(data = departmentwise) +
+   geom_point(aes(x = Gender, y = Admitted, colour = Dept))
> # or
>
> qplot(Gender, Admitted, data = departmentwise, colour = Dept)
```



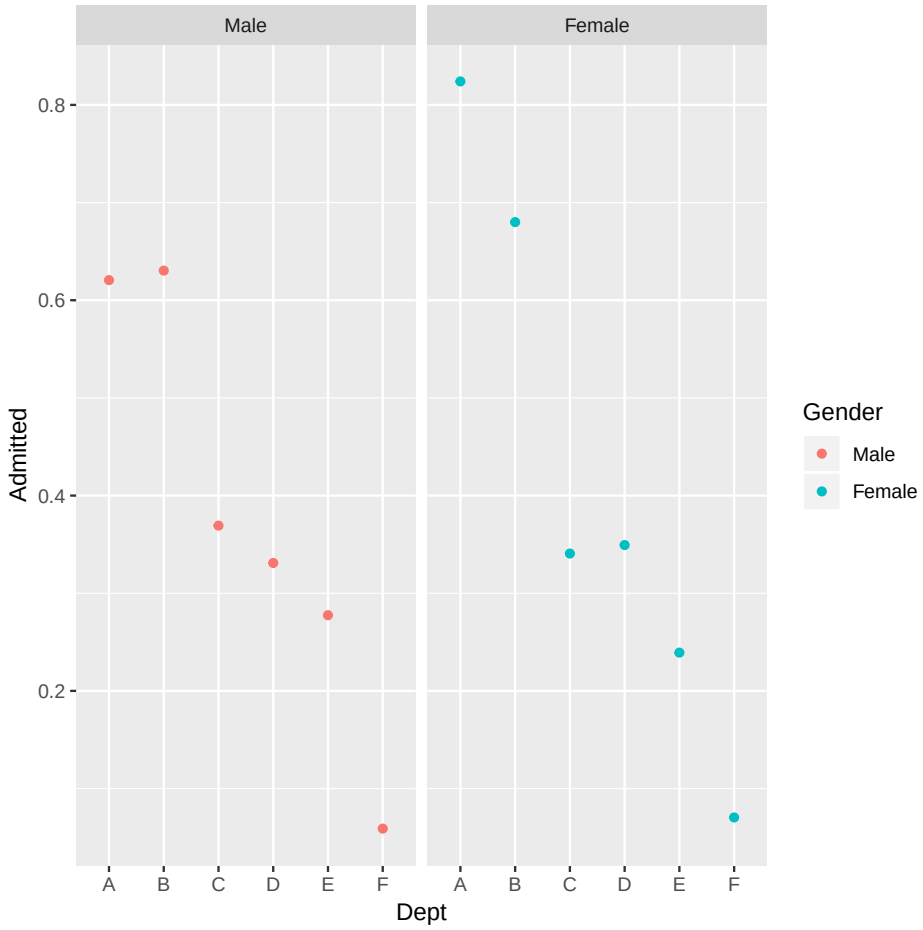

```
> ggplot(data = departmentwise) +
+   geom_point(aes(x = Gender, y = Admitted, colour = Dept)) +
+   facet_grid(~ Dept)
> # or
>
> qplot(Gender, Admitted, data = departmentwise, colour = Dept) +
+   facet_grid(~ Dept)
```



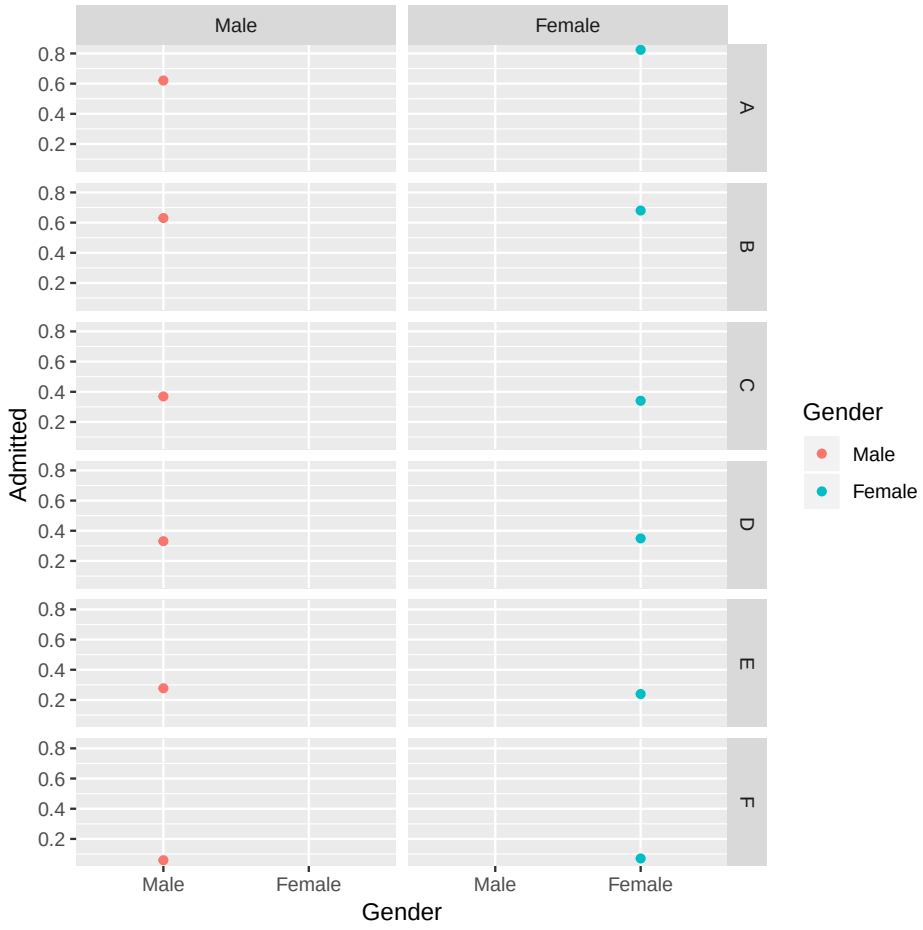

```
> ggplot(data = departmentwise) +
+   geom_point(aes(x = Gender, y = Admitted, colour = Gender)) +
+   facet_grid(~ Dept)
> # or
>
> qplot(Gender, Admitted, data = departmentwise, colour = Gender) +
+   facet_grid(~ Dept)
```



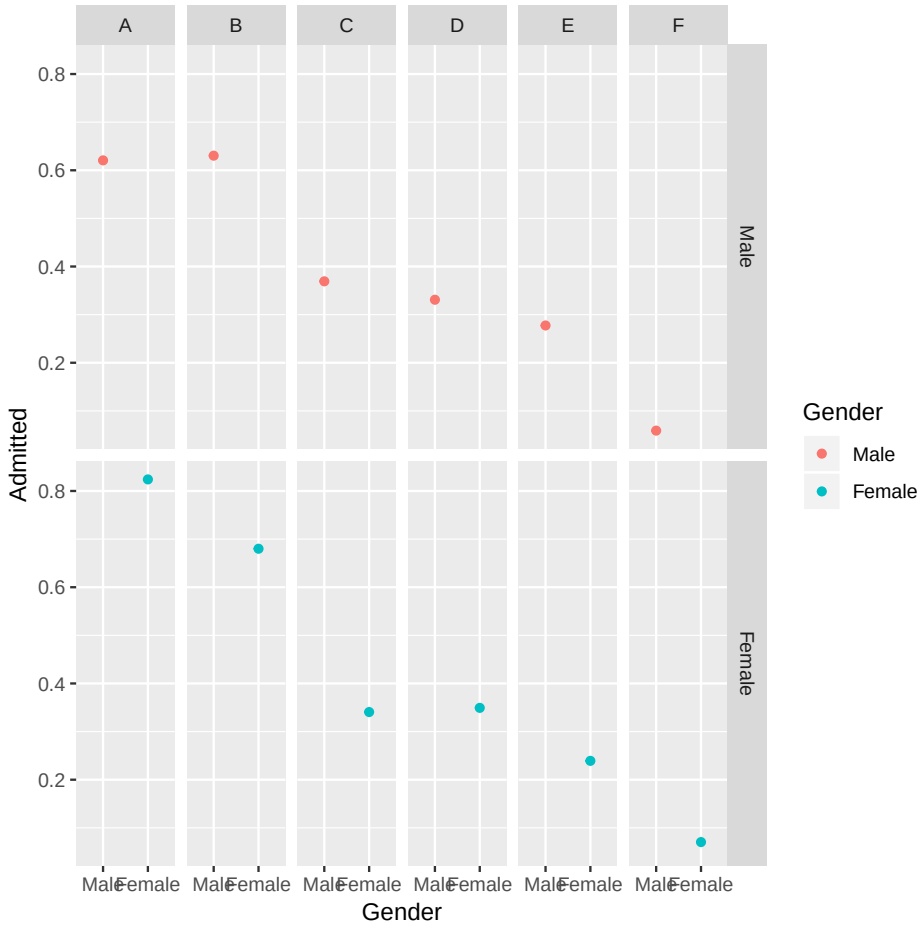

```
> ggplot(data = departmentwise) +
+   geom_point(aes(x = Dept, y = Admitted, colour = Gender)) +
+   facet_grid(~ Gender)
> # or
>
> qplot(Dept, Admitted, data = departmentwise, colour = Gender) +
+   facet_grid(~ Gender)
```



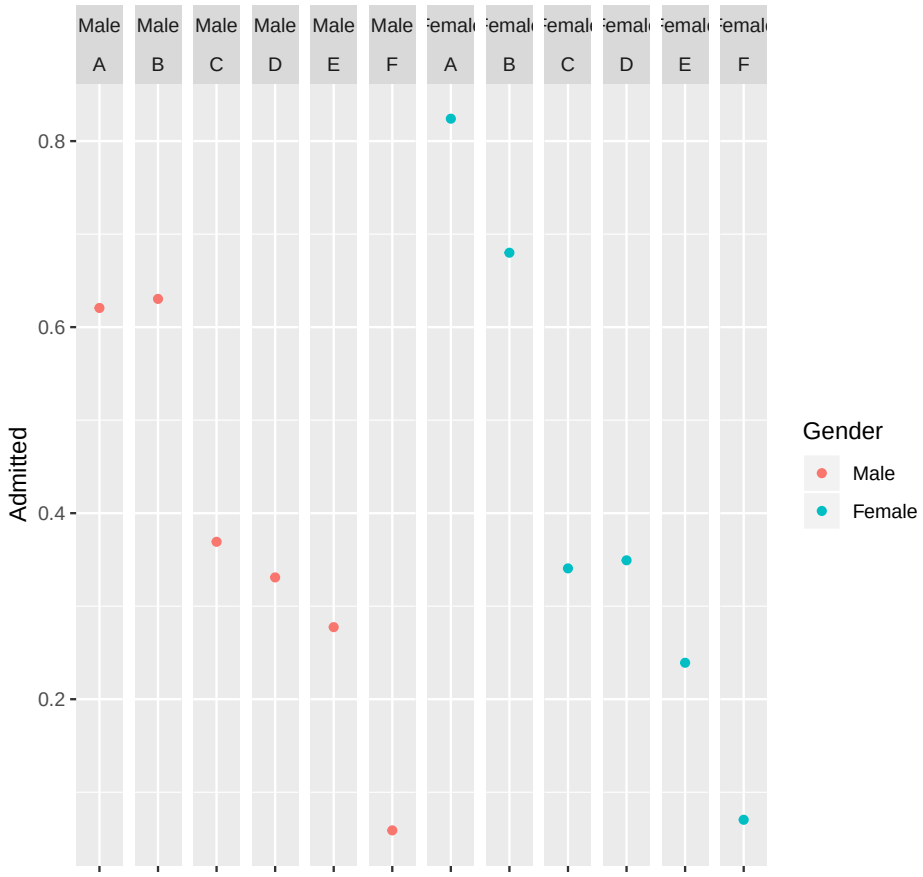

```
> ggplot(data = departmentwise) +
+   geom_point(aes(x = Gender, y = Admitted, colour = Gender)) +
+   facet_grid(Dept ~ Gender)
> # or
>
> qplot(Gender, Admitted, data = departmentwise, colour = Gender) +
+   facet_grid(Dept ~ Gender)
```



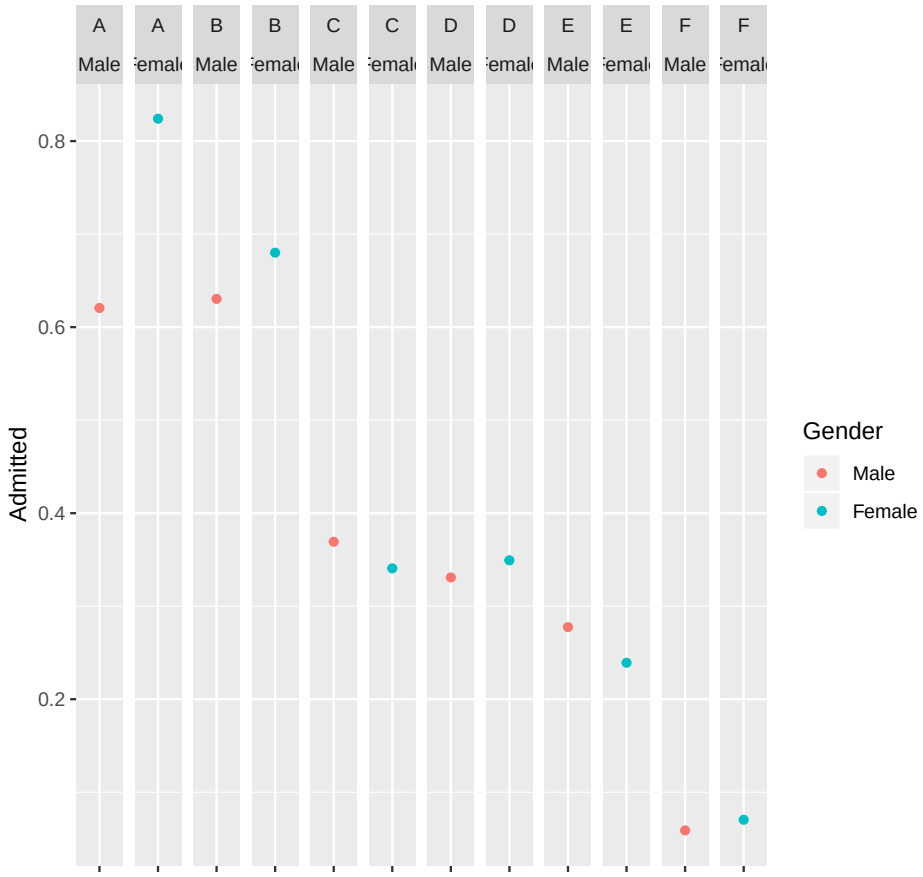

```
> qplot(Gender, Admitted, data = departmentwise, colour = Gender) +
+   facet_grid(Gender ~ Dept)
```




```
> qplot("", Admitted, data = departmentwise, colour = Gender) +
+   facet_grid(~ Gender + Dept)
```




```
> qplot("", Admitted, data = departmentwise, colour = Gender) +
+   facet_grid(~ Dept + Gender)
```



Note:

We will revisit dot plots in future chapters in Statistical Visualization I & II. If you are interested in more sophisticated ones at this time, see this blog from Nina Zumel where she creates several of the graphs from **?** via ggplot2:

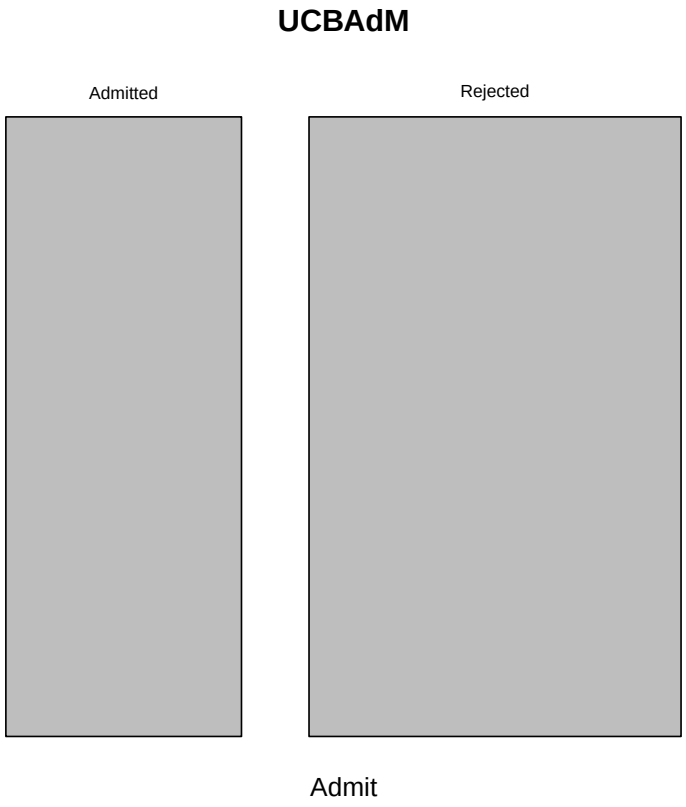
`http://www.win-vector.com/blog/2013/02/revisiting-clevelands-the-elements-of-graphing-da`

1.2.4 Mosaic Plots

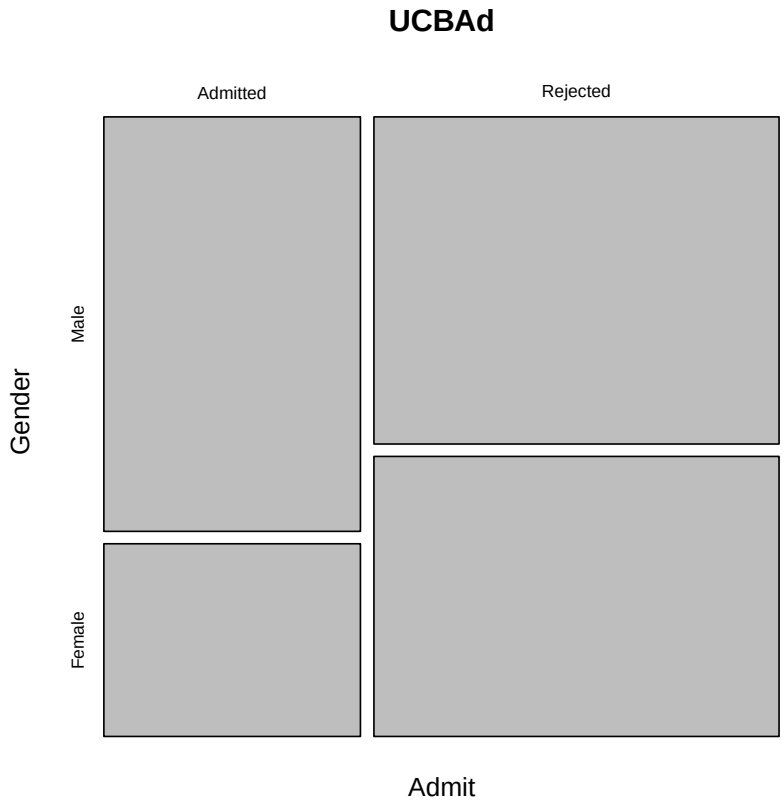
The R help page for mosaicplot indicates:

> "Plots a mosaic on the current graphics device. [...]
> *shade* a logical indicating whether to produce extended mosaic plots, or a numeric vector of at most 5 distinct positive numbers giving the absolute values of the cut points for the residuals. By default, shade is FALSE, and simple mosaics are created. Using shade = TRUE cuts absolute values at 2 and 4."

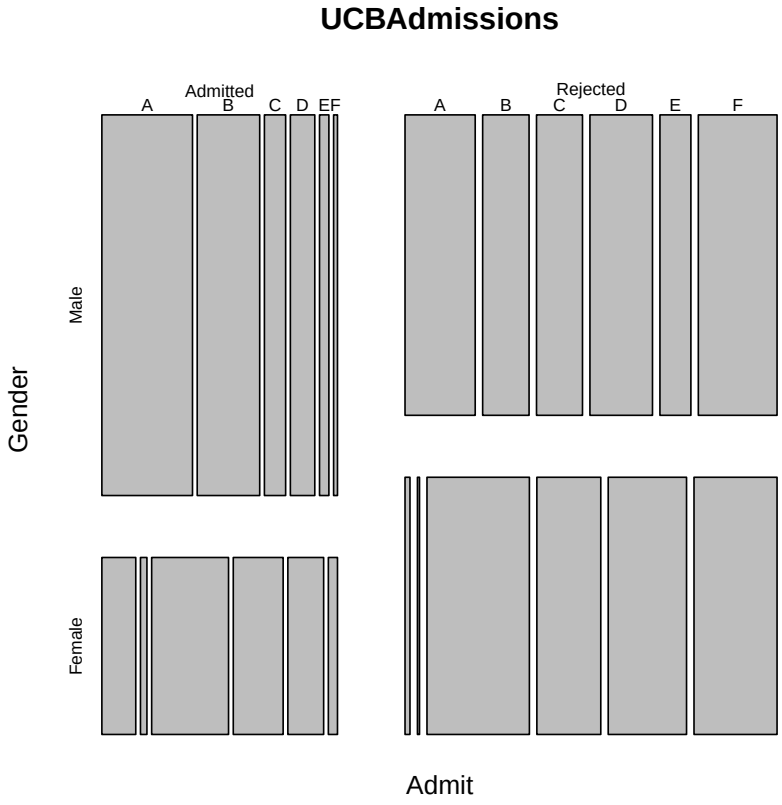
```
> UCBAdM <- margin.table(UCBAdmissions, 1)
> UCBAdM

Admit
Admitted Rejected
    1755     2771

> mosaicplot(UCBAdM)
```




```
> UCBAd <- margin.table(UCBAdmissions, 1:2)
> UCBAd

          Gender
Admit      Male Female
  Admitted 1198    557
  Rejected 1493   1278

> mosaicplot(UCBAd)
```




```
> mosaicplot(UCBAdmissions)
```

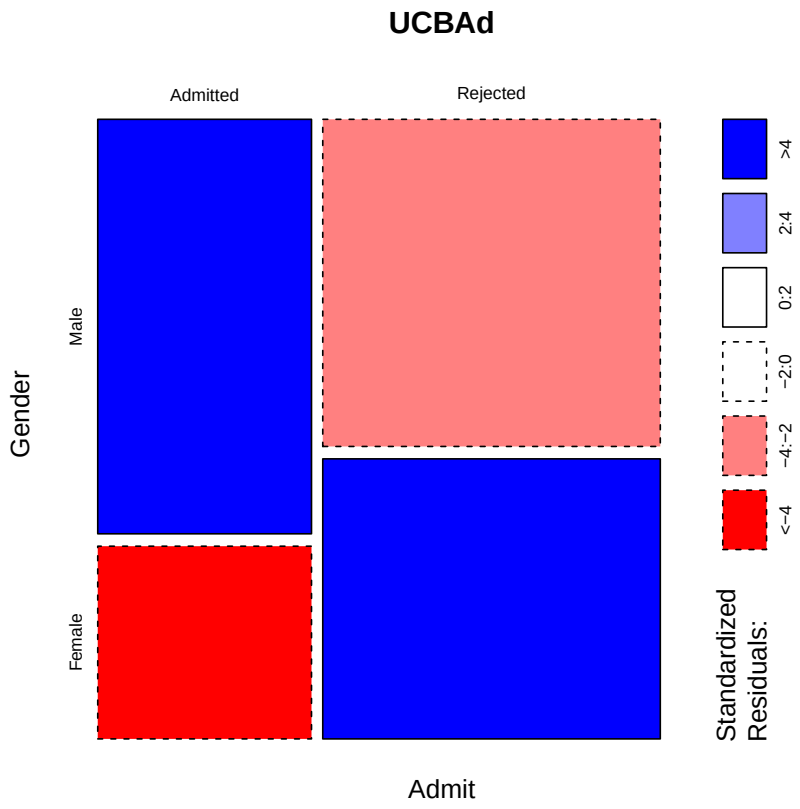


Additional argument *shade*:

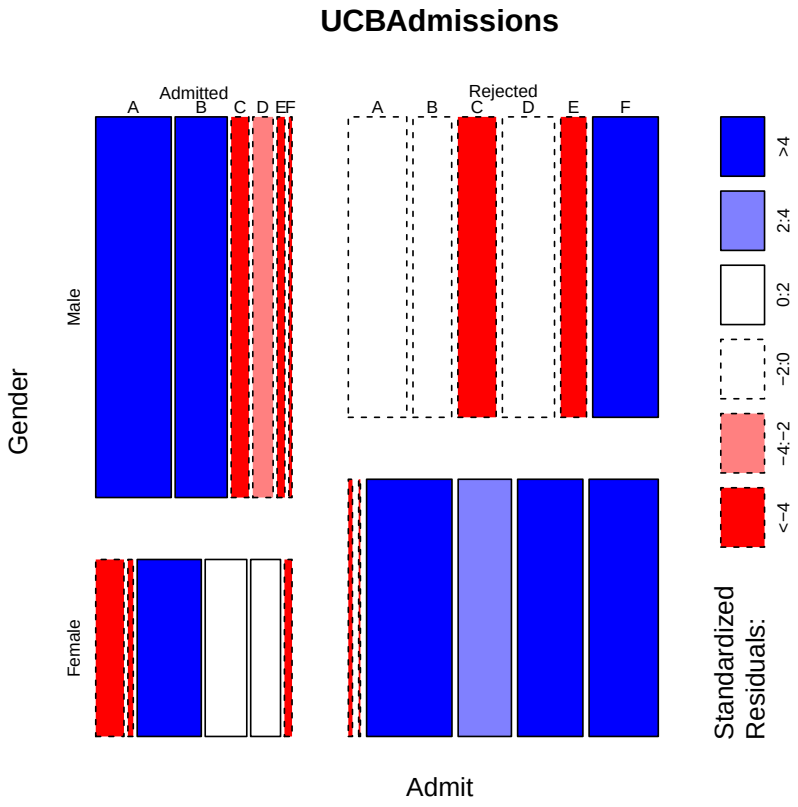
> "a logical indicating whether to produce extended mosaic plots, or a numeric vector of at most 5 distinct positive numbers giving the absolute values of the cut points for the residuals. By default, shade is FALSE, and simple mosaics are created. Using shade = TRUE cuts absolute values at 2 and 4.
> Extended mosaic displays visualize standardized residuals of a loglinear model for the table by color and outline of the mosaic's tiles. (Standardized residuals are often referred to a standard normal distribution.) Cells representing negative residuals are drawn in shaded of red and with broken borders; positive ones are drawn in blue with solid borders."

Just using "shade = TRUE" assumes an independence model of the variables in the mosaic plot.

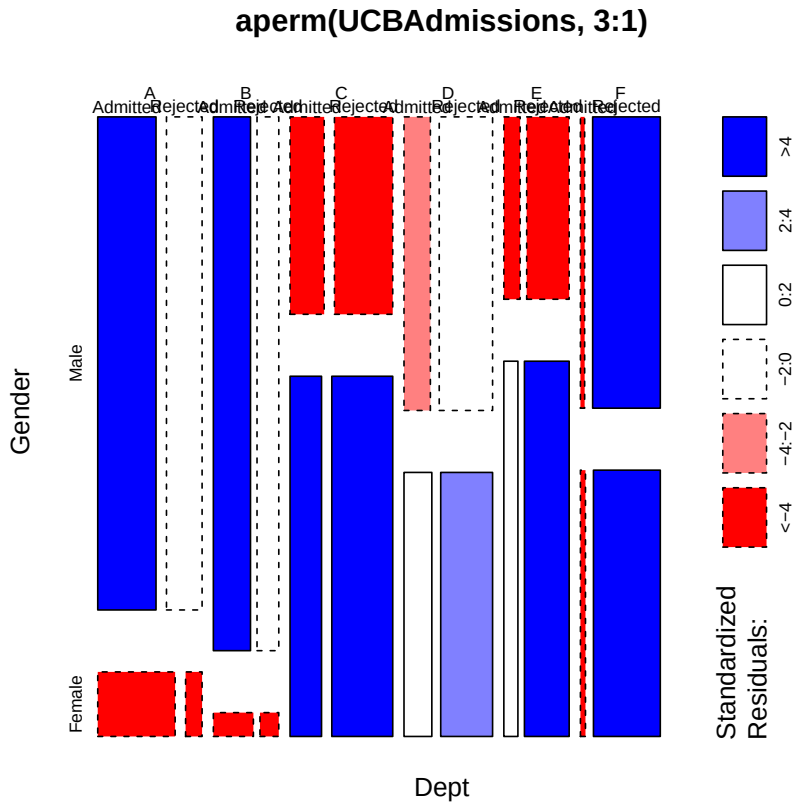
```
> mosaicplot(UCBAd, shade = TRUE)
```



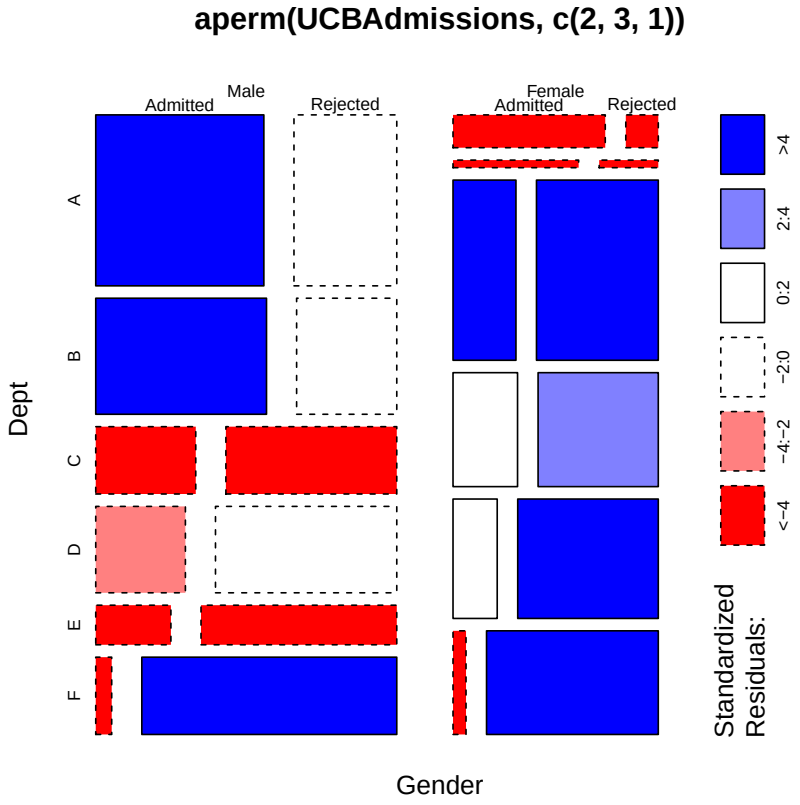

```
> mosaicplot(UCBAdmissions, shade = TRUE)
```




```
> mosaicplot(aperm(UCBAdmissions, 3:1), shade = TRUE)
```



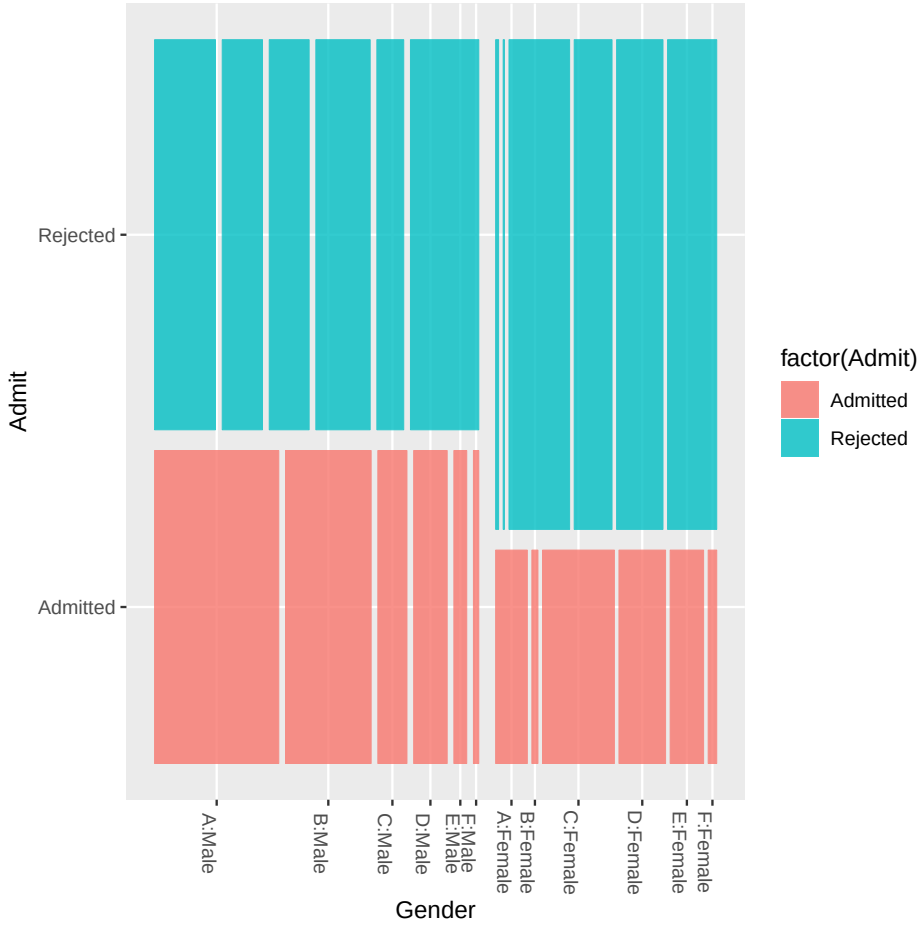

```
> mosaicplot(aperm(UCBAdmissions, c(2, 3, 1)), shade = TRUE)
```



Mosaic plots in ggplot2 are produced via the *ggmosaic* package. See `https://cran.r-project.org/web/packages/ggmosaic/vignettes/ggmosaic.html` for further details:

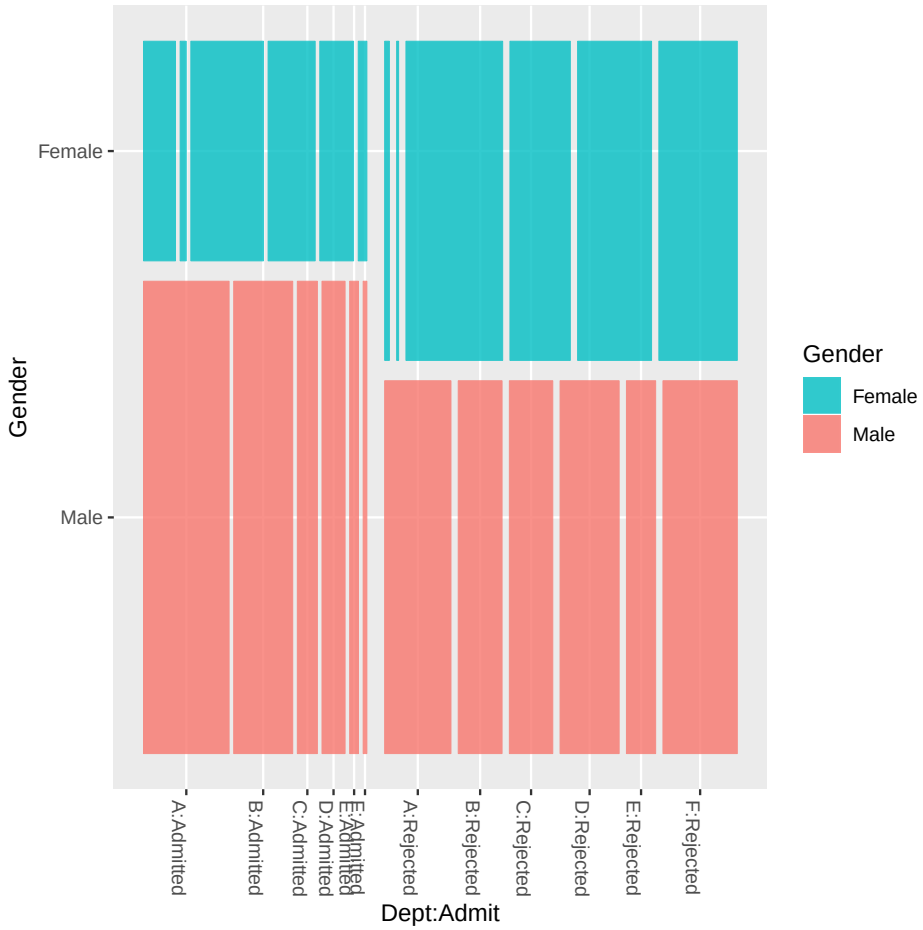
```
> library(ggmosaic)
> library(tibble)
> # function countsToCases taken from
> # http://www.cookbook-r.com/Manipulating_data/Converting_between_data_frames_and_con
>
> countsToCases <- function(x, countcol = "Freq") {
+     # Get the row indices to pull from x
+     idx <- rep.int(seq_len(nrow(x)), x[[countcol]])
+
+     # Drop count column
+     x[[countcol]] <- NULL
```

```
+
+     # Get the rows from x
+     x[idx, ]
+ }
> UCBAdmissionstib <- as.tibble(countsToCases(as.data.frame(UCBAdmissions)))
> ggplot(data = UCBAdmissionstib) +
+    geom_mosaic(aes(x = product(Dept, Admit, Gender), fill = factor(Admit)),
+                divider = mosaic("h"), offset = 0.03) +
+   labs(x = "Gender", y = "Admit") +
+   theme(axis.text.x = element_text(angle = -90, hjust = .1))
```




```
> ggplot(data = UCBAdmissionstib) +
+    geom_mosaic(aes(x = product(Dept, Gender, Admit), fill = factor(Gender)),
+                divider = mosaic("h"), offset = 0.03) +
+   labs(x = "Dept:Admit", y = "Gender") +
```

```
+    theme(axis.text.x = element_text(angle = -90, hjust = .1)) +
+    guides(fill = guide_legend(title = "Gender", reverse = TRUE))
```

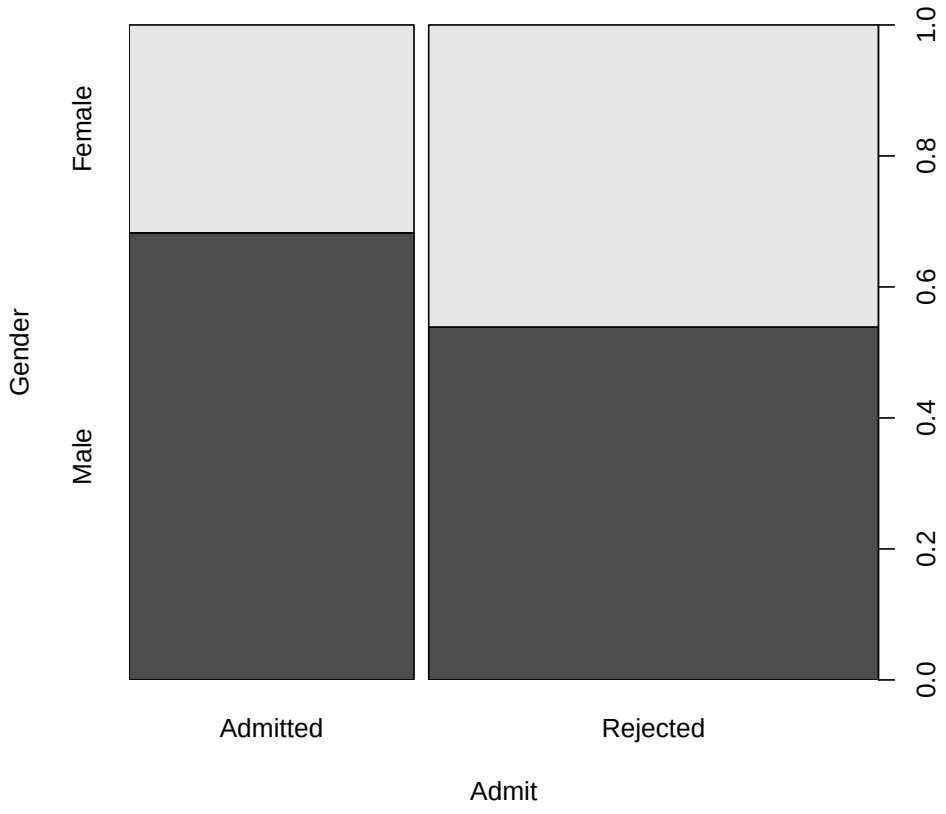


1.2.5 Spine Plots and Spinograms

The R help page for spineplot indicates:

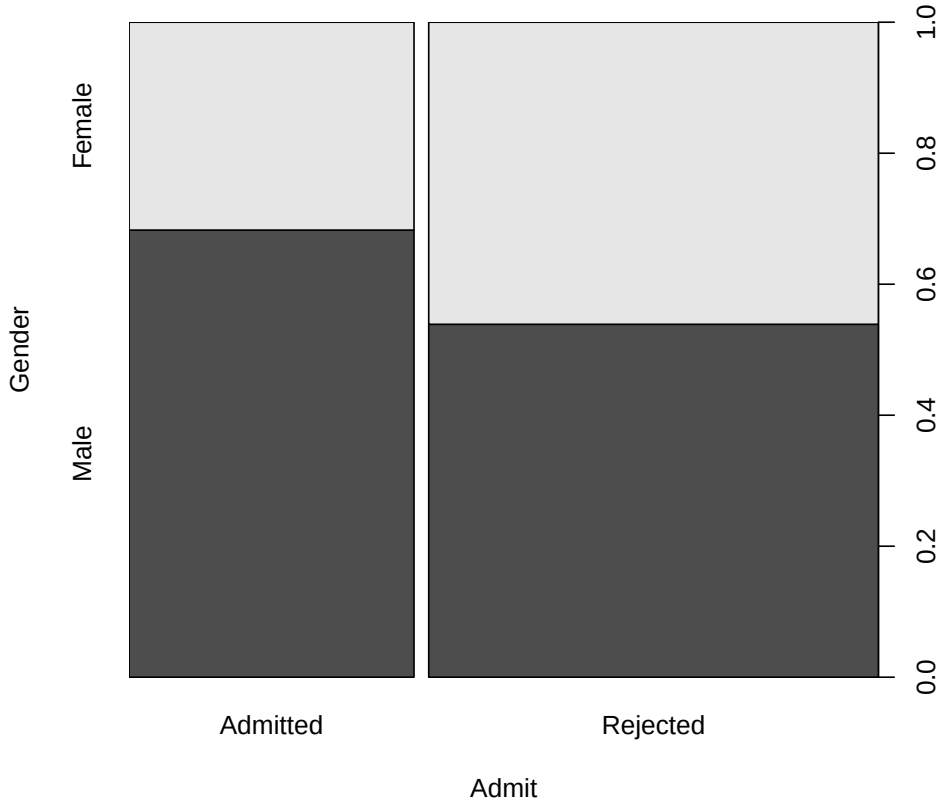
> "Spine plots are a special case of mosaic plots, and can be seen as a generalization of stacked (or highlighted) bar plots. Analogously, spinograms are an extension of histograms."

```
> #
> # compare the use of this command without () ...
> #
> spineplot(UCBAd)
```



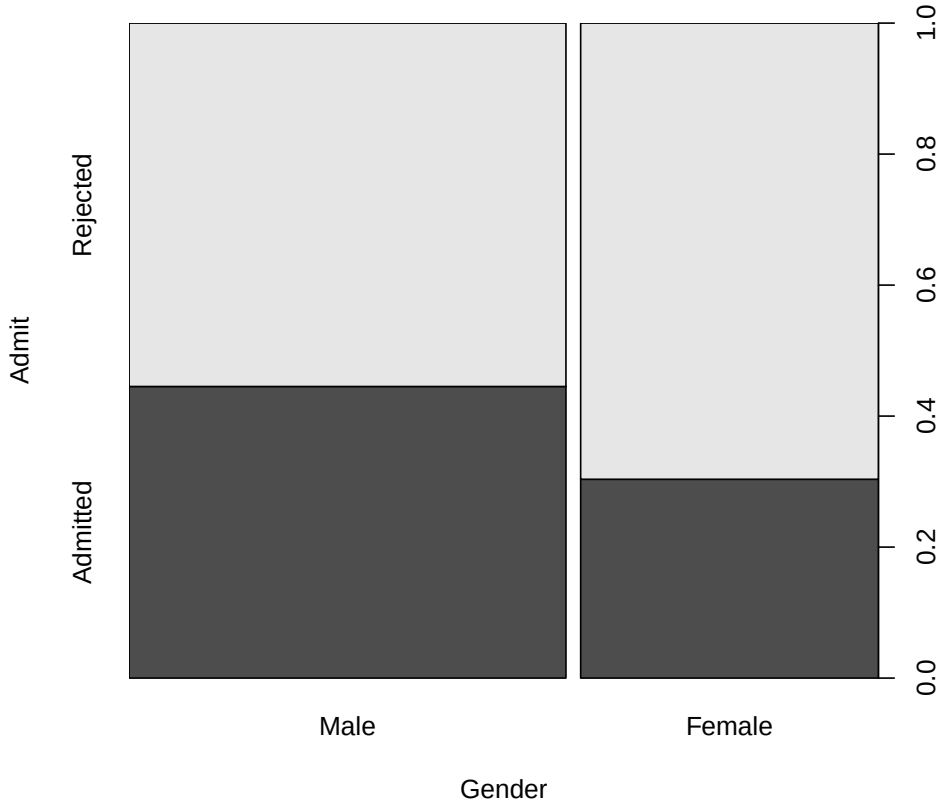

```
> #
> # ... and with ()
> #
> (spineplot(UCBAd))
```

```
          Gender
Admit      Male Female
  Admitted 1198    557
  Rejected 1493   1278
```



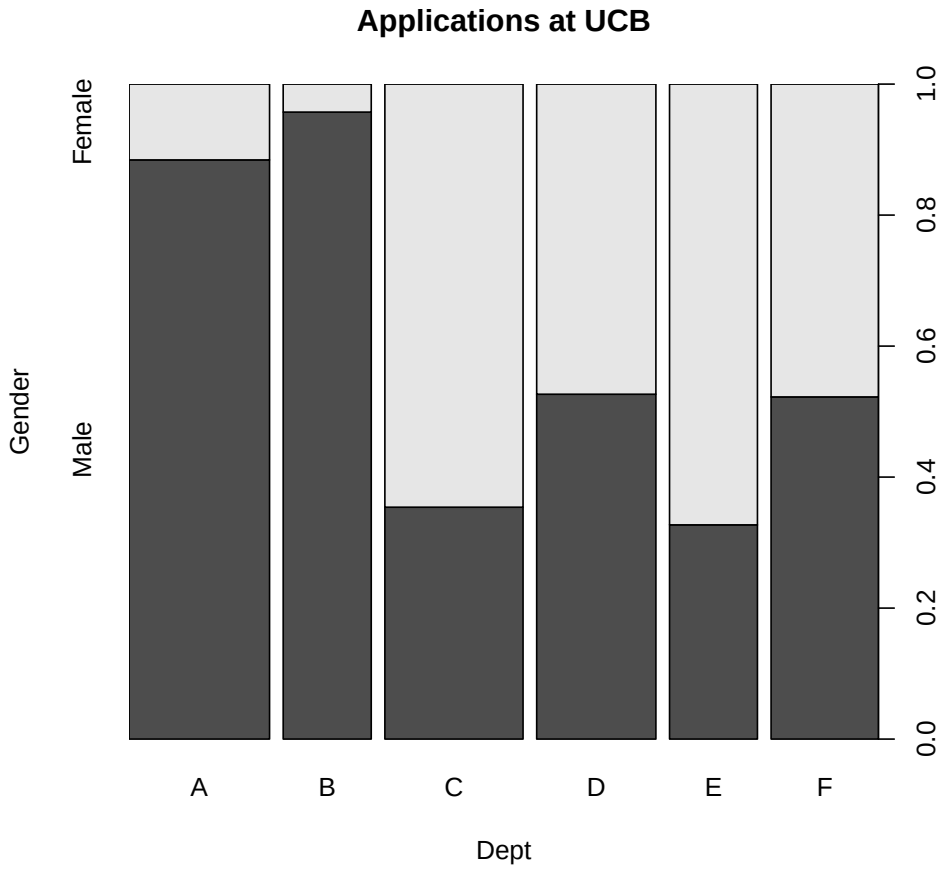

```
> (spineplot(t(UCBAd)))
```

```
        Admit
Gender   Admitted Rejected
  Male       1198     1493
  Female      557     1278
```



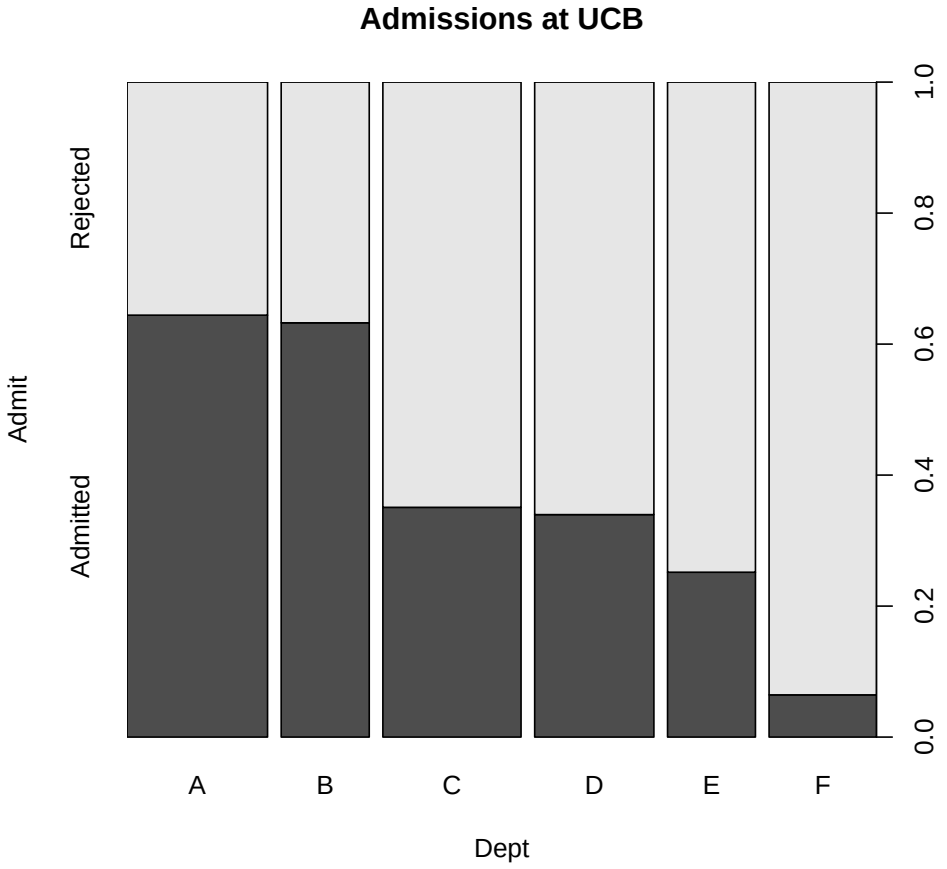

```
> (spineplot(margin.table(UCBAdmissions, c(3, 2)), main = "Applications at UCB"))
```

```
    Gender
Dept Male Female
   A  825    108
   B  560     25
   C  325    593
   D  417    375
   E  191    393
   F  373    341
```




```
> (spineplot(margin.table(UCBAdmissions, c(3, 1)), main = "Admissions at UCB"))
```

```
    Admit
Dept Admitted Rejected
   A      601      332
   B      370      215
   C      322      596
   D      269      523
   E      147      437
   F       46      668
```


Admissions at UCB
Admit
Rejected
Admitted
0.0
0.2
0.4
0.6
0.8
1.0
A
B
C
D
E
F
Dept

1.3 Categorical Plots in Mondrian

According to `http://www.theusrus.de/Mondrian/`:

> "Mondrian is a general purpose statistical data–visualization system. It features outstanding visualization techniques for data of almost any kind, and has its particular strength compared to other tools when working with **Categorical Data, Geographical Data** and **LARGE Data**.
>
> All plots in Mondrian are fully linked, and offer various interactions and queries. Any case selected in a plot in Mondrian is highlighted in all other plots.
>
> Currently implemented plots comprise **Mosaic Plot, Scatterplots and SPLOM, Maps, Barcharts, Histograms, Missing Value Plot, Parallel Coordinates/Boxplots** and **Boxplots y by x**. "

Main references for Mondrian are **?**, **?**, and **?**. **?** has an associated Web page at `http://www.interactivegraphics.org`:

> "This site is the web resource for the book "Interactive Graphics for Data Analysis — Principles and Examples".
>
> There are links to the most important software tools, all datasets used in the book for easy download, and a set of slides which may be used together with the book for a lecture.
>
> The R–code used in the book can be found here as well."

1.3.1 Installation

Go to `http://www.theusrus.de/Mondrian/`, then follow the link to the download section on this page. Look over the license condition. If you agree, then download the most recent version of Mondrian (currently 1.5b as of 8/28/2013) by right mouse–clicking on the operating system you use. Save *Mondrian.exe* into a directory of your choice. You can start Mondrian directly (without any additional installation) by mouse–clicking on *Mondrian.exe*.

As a test data set, work with the *Titanic* data available under the *Mondrian Titanic* link or directly from `http://www.theusrus.de/Mondrian/Data/Titanic.txt`. Save these data locally as *Titanic.txt*. Then load them into *Mondrian*.

1.3.2 The Titanic Data in Mondrian

The *Mondrian* description at `http://www.theusrus.de/Mondrian/` indicates:

> "**Titanic**
>
> Data set on the 2201 passengers of the Titanic. Pure categorical with data on class, gender, age and survival."

The interactive exploration of the *Titanic* data via *Mondrian* has been further discussed in **?**, Examples D: The Titanic Disaster Revisited, pp. 183–191.

Task:
Interactively recreate the nine plots from Figure **??** using *Mondrian*.

Answer:

These plots were created via the following settings:

(1) Survived: Bar Chart Yes	**(5) Age:** Bar Chart
(2) Class: Bar Chart Sort By: Relative Selected	**(6) Sex:** Bar Chart
(3) Class: Bar Chart Spine Plot	**(7) Age** Bar Chart Spine Plot
(4) Sex & Class: Mosaic Plot	**(8) Sex:** Bar Chart Spine Plot
	(9) Class & Age & Sex: Mosaic Plot Display As: Same Bin Size

1.3.3 The Titanic Data in iPlots

iPlots is an interactive graphics package for R, further described in **?** and accessible at `http://cran.r-project.org/web/packages/iplots/index.html`. *iPlots* is closely related to *Mondrian*. In the example below, we will reproduce some of the plots via *iPlots* that were previously created via *Mondrian*.

```
> Sys.setenv(JAVA_HOME = "C:/Program Files/Java/jdk-11.0.1/")
> library(iplots)
> TitanicMondrian <- read.table("http://www.theusrus.de/Mondrian/Data/Titanic.txt",
+                               header = TRUE)
> summary(TitanicMondrian)

    Class        Age           Sex       Survived
 Crew  :885   Adult:2092   Female: 470   No :1490
 First :325   Child: 109   Male  :1731   Yes: 711
 Second:285
 Third :706

> ibar(TitanicMondrian$Survived)

ID:1 Name: "Barchart (TitanicMondrian$Survived)"

> iplot.rotate(1)
> iset.select(TitanicMondrian$Survived == "Yes")
> ibar(TitanicMondrian$Class)

ID:2 Name: "Barchart (TitanicMondrian$Class)"

> iplot.rotate(1)
> ibar(TitanicMondrian$Class, isSpine = TRUE)

ID:3 Name: "Barchart (TitanicMondrian$Class)"

> iplot.rotate(1)
> ibar(TitanicMondrian$Age)
```

```
ID:4 Name: "Barchart (TitanicMondrian$Age)"

> iplot.rotate(1)
> imosaic(TitanicMondrian$Sex, TitanicMondrian$Class)

ID:5 Name: "Mosaic plot (TitanicMondrian$Sex, TitanicMondrian$Class)"

> imosaic(TitanicMondrian$Sex, TitanicMondrian$Class, TitanicMondrian$Age)

ID:6 Name: "Mosaic plot (TitanicMondrian$Sex, TitanicMondrian$Class, TitanicMondrian$Age

> iset.selectNone()
> iobj.list()

[[1]]
PlotText(labels=2,coord=0/0,visible=true)

[[2]]
PlotText(labels=4,coord=0/0,visible=true)

> for (i in 1:length(iobj.list()))
+   iobj.rm() # remove all objects
> iplot.list()

[[1]]
ID:1 Name: "Barchart (TitanicMondrian$Survived)"

[[2]]
ID:2 Name: "Barchart (TitanicMondrian$Class)"

[[3]]
ID:3 Name: "Barchart (TitanicMondrian$Class)"

[[4]]
ID:4 Name: "Barchart (TitanicMondrian$Age)"
```

```
[[5]]
ID:5 Name: "Mosaic plot (TitanicMondrian$Sex, TitanicMondrian$Class)"

[[6]]
ID:6 Name: "Mosaic plot (TitanicMondrian$Sex, TitanicMondrian$Class, TitanicMondrian$Age

> for (i in 1:length(iplot.list()))
+   iplot.off() # close all windows
>
>
> #iset.rm() # remove this iset
```

1.4 Further Reading

Additional sources for the visualization of categorical data are:

- ?
- ?
- ?
- ?

— THE END —