Chapter 8

Statistical Models

A statistical model describes the relationship between one or more explanatory variables and one or more response variables. Graphs can help to visualize these relationships. In this section we'll focus on models that have a single response variable that is either quantitative (a number) or binary (yes/no).

8.1 Correlation plots

Correlation plots help you to visualize the pairwise relationships between a set of quantitative variables by displaying their correlations using color or shading.

Consider the Saratoga Houses dataset, which contains the sale price and characteristics of Saratoga County, NY homes in 2006. In order to explore the relationships among the quantitative variables, we can calculate the Pearson Product-Moment correlation coefficients.

```
data(SaratogaHouses, package="mosaicData")

# select numeric variables
df <- dplyr::select_if(SaratogaHouses, is.numeric)

# calulate the correlations
r <- cor(df, use="complete.obs")
round(r,2)
```

```
##            price lotSize   age landValue livingArea pctCollege bedrooms
## price       1.00    0.16 -0.19      0.58       0.71       0.20     0.40
## lotSize     0.16    1.00 -0.02      0.06       0.16      -0.03     0.11
## age        -0.19   -0.02  1.00     -0.02      -0.17      -0.04     0.03
## landValue   0.58    0.06 -0.02      1.00       0.42       0.23     0.20
## livingArea  0.71    0.16 -0.17      0.42       1.00       0.21     0.66
## pctCollege  0.20   -0.03 -0.04      0.23       0.21       1.00     0.16
## bedrooms    0.40    0.11  0.03      0.20       0.66       0.16     1.00
## fireplaces  0.38    0.09 -0.17      0.21       0.47       0.25     0.28
## bathrooms   0.60    0.08 -0.36      0.30       0.72       0.18     0.46
## rooms       0.53    0.14 -0.08      0.30       0.73       0.16     0.67
##            fireplaces bathrooms rooms
## price            0.38      0.60  0.53
## lotSize          0.09      0.08  0.14
```

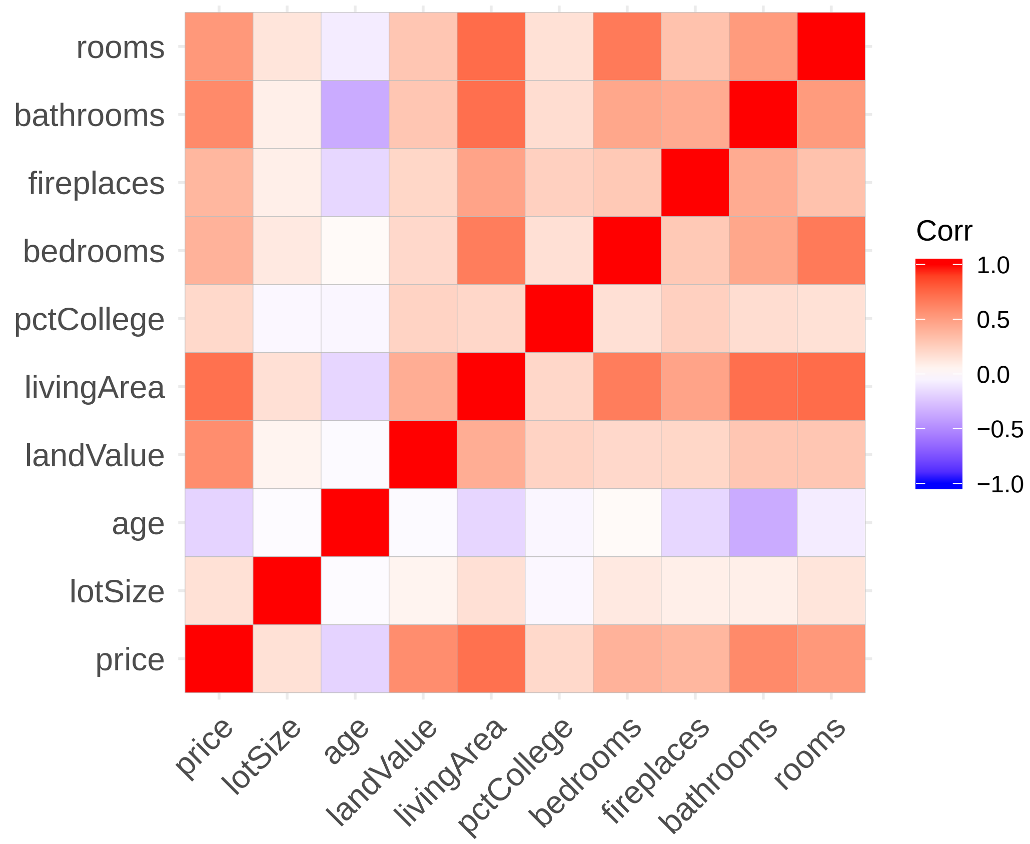


Figure 8.1: Correlation matrix

```
## age              -0.17     -0.36 -0.08
## landValue         0.21      0.30  0.30
## livingArea        0.47      0.72  0.73
## pctCollege        0.25      0.18  0.16
## bedrooms          0.28      0.46  0.67
## fireplaces        1.00      0.44  0.32
## bathrooms         0.44      1.00  0.52
## rooms             0.32      0.52  1.00
```

The `ggcorrplot` function in the `ggcorrplot` package can be used to visualize these correlations. By default, it creates a `ggplot2` graph were darker red indicates stronger positive correlations, darker blue indicates stronger negative correlations and white indicates no correlation.

```
library(ggplot2)
library(ggcorrplot)
ggcorrplot(r)
```

From the graph, an increase in number of bathrooms and living area are associated with increased price, while older homes tend to be less expensive. Older homes also tend to have fewer bathrooms.

The `ggcorrplot` function has a number of options for customizing the output. For example

- `hc.order = TRUE` reorders the variables, placing variables with similar correlation patterns together.

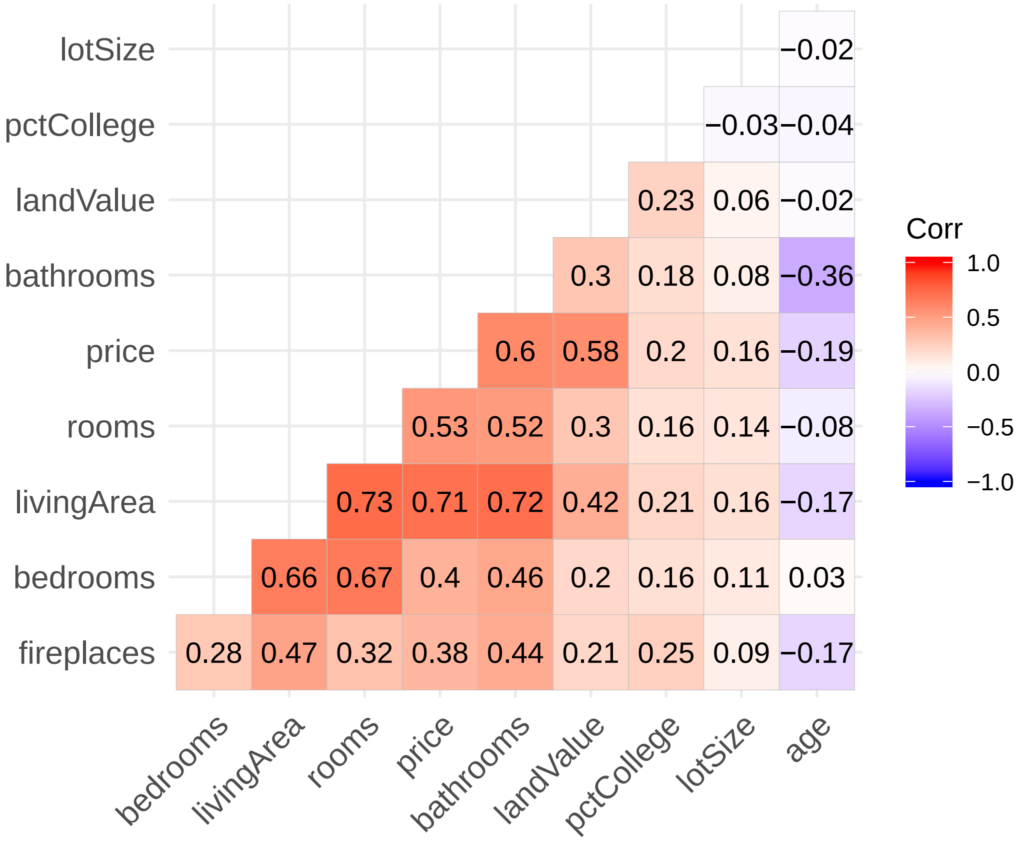


Figure 8.2: Sorted lower triangel correlation matrix with options

- `type = "lower"` plots the lower portion of the correlation matrix.
- `lab = TRUE` overlays the correlation coefficients (as text) on the plot.

```
ggcorrplot(r,
           hc.order = TRUE,
           type = "lower",
           lab = TRUE)
```

These, and other options, can make the graph easier to read and interpret.

8.2 Linear Regression

Linear regression allows us to explore the relationship between a quantitative response variable and an explanatory variable while other variables are held constant.

Consider the prediction of home prices in the Saratoga dataset from lot size (square feet), age (years), land value (1000s dollars), living area (square feet), number of bedrooms and bathrooms and whether the home is on the waterfront or not.

```
data(SaratogaHouses, package="mosaicData")
houses_lm <- lm(price ~ lotSize + age + landValue +
                  livingArea + bedrooms + bathrooms +
```

Table 8.1: Linear Regression results

term	estimate	std.error	statistic	p.value
(Intercept)	139878.80	16472.93	8.49	0.00
lotSize	7500.79	2075.14	3.61	0.00
age	-136.04	54.16	-2.51	0.01
landValue	0.91	0.05	19.84	0.00
livingArea	75.18	4.16	18.08	0.00
bedrooms	-5766.76	2388.43	-2.41	0.02
bathrooms	24547.11	3332.27	7.37	0.00
waterfrontNo	-120726.62	15600.83	-7.74	0.00

```
                  waterfront,
                data = SaratogaHouses)
```

From the results, we can estimate that an increase of one square foot of living area is associated with a home price increase of $75, holding the other variables constant. Additionally, waterfront home cost approximately $120,726 more than non-waterfront home, again controlling for the other variables in the model.

The visreg package provides tools for visualizing these conditional relationships.

The `visreg` function takes (1) the model and (2) the variable of interest and plots the conditional relationship, controlling for the other variables. The option `gg = TRUE` is used to produce a `ggplot2` graph.

```
# conditional plot of price vs. living area
library(ggplot2)
library(visreg)
visreg(houses_lm, "livingArea", gg = TRUE)
```

The graph suggests that, after controlling for lot size, age, living area, number of bedrooms and bathrooms, and waterfront location, sales price increases with living area in a linear fashion.

> **How does `visreg` work?** The fitted model is used to predict values of the response variable, across the range of the chosen explanatory variable. The other variables are set to their median value (for numeric variables) or most frequent category (for categorical variables). The user can override these defaults and chose specific values for any variable in the model.

Continuing the example, the price difference between waterfront and non-waterfront homes is plotted, controlling for the other seven variables. Since a `ggplot2` graph is produced, other `ggplot2` functions can be added to customize the graph.

```
# conditional plot of price vs. waterfront location
visreg(houses_lm, "waterfront", gg = TRUE) +
  scale_y_continuous(label = scales::dollar) +
  labs(title = "Relationship between price and location",
       subtitle = "controlling for lot size, age, land value, bedrooms and bathrooms",
       caption = "source: Saratoga Housing Data (2006)",
       y = "Home Price",
       x = "Waterfront")
```

There are far fewer homes on the water, and they tend to be more expensive (even controlling for size, age, and land value).

The `vizreg` package provides a wide range of plotting capabilities. See Visualization of regression models using visreg for details.

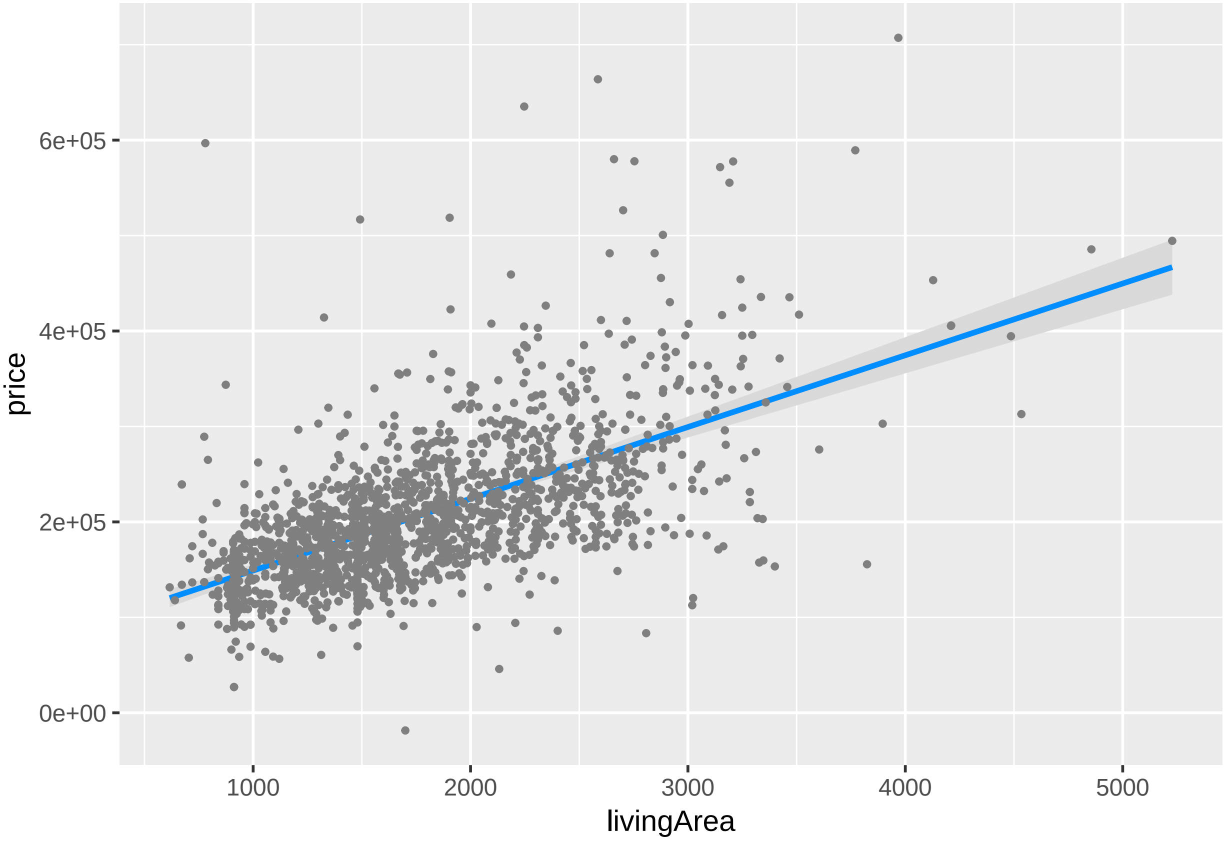


Figure 8.3: Conditional plot of living area and price

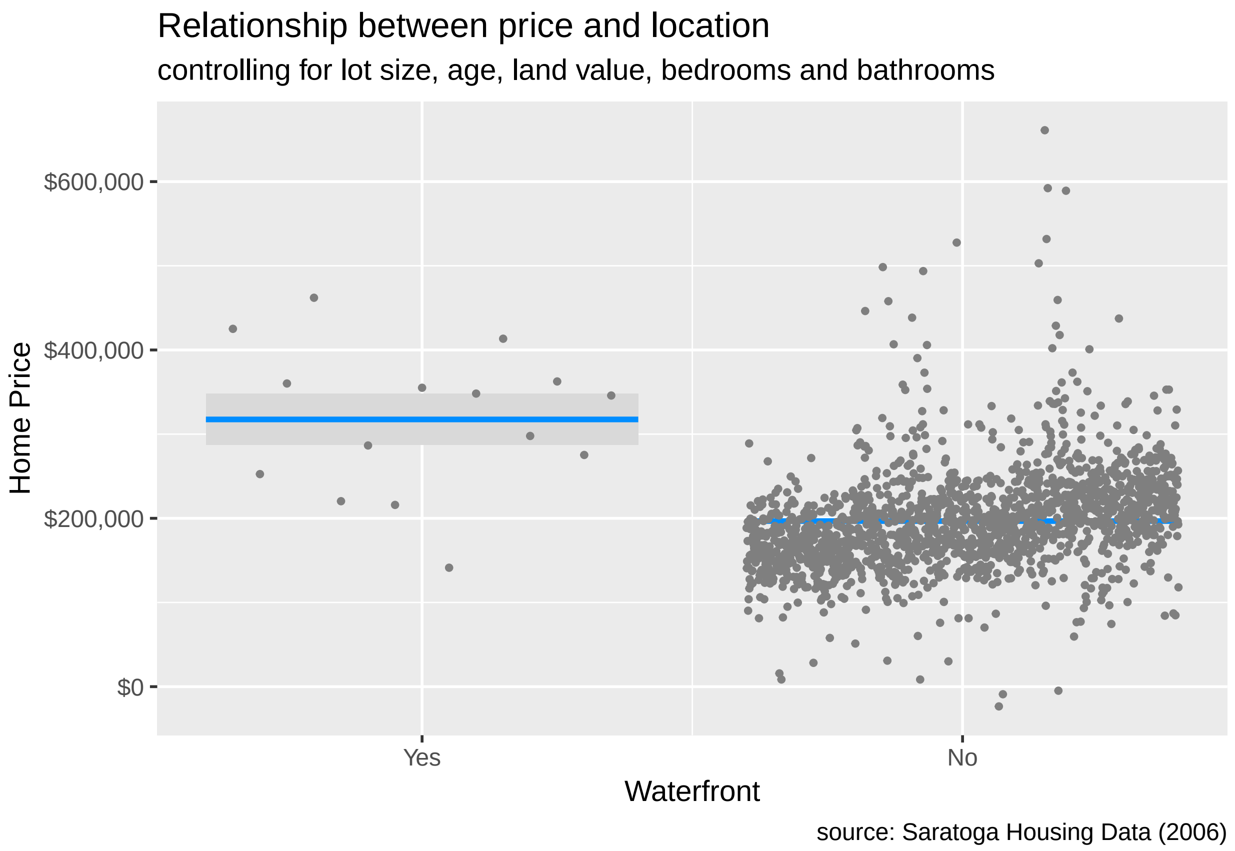


Figure 8.4: Conditional plot of location and price

8.3 Logistic regression

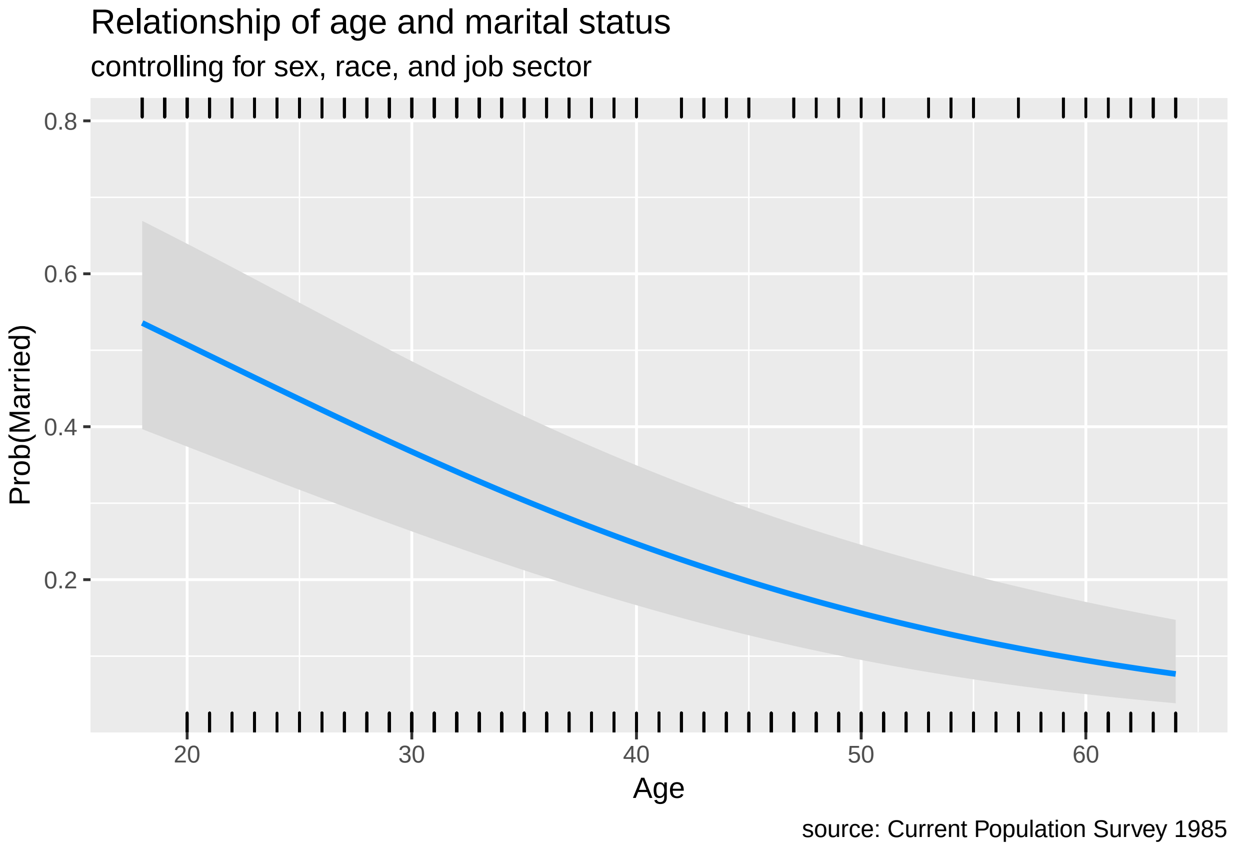
Logistic regression can be used to explore the relationship between a binary response variable and an explanatory variable while other variables are held constant. Binary response variables have two levels (yes/no, lived/died, pass/fail, malignant/benign). As with linear regression, we can use the visreg package to visualize these relationships.

Using the CPS85 data let's predict the log-odds of being married, given one's sex, age, race and job sector.

```r
# fit logistic model for predicting
# marital status: married/single
data(CPS85, package = "mosaicData")
cps85_glm <- glm(married ~ sex + age + race + sector,
                 family="binomial",
                 data=CPS85)
```

Using the fitted model, let's visualize the relationship between age and the probability of being married, holding the other variables constant. Again, the `visreg` function takes the model and the variable of interest and plots the conditional relationship, controlling for the other variables. The option `gg = TRUE` is used to produce a `ggplot2` graph. The `scale = "response"` option creates a plot based on a probability (rather than log-odds) scale.

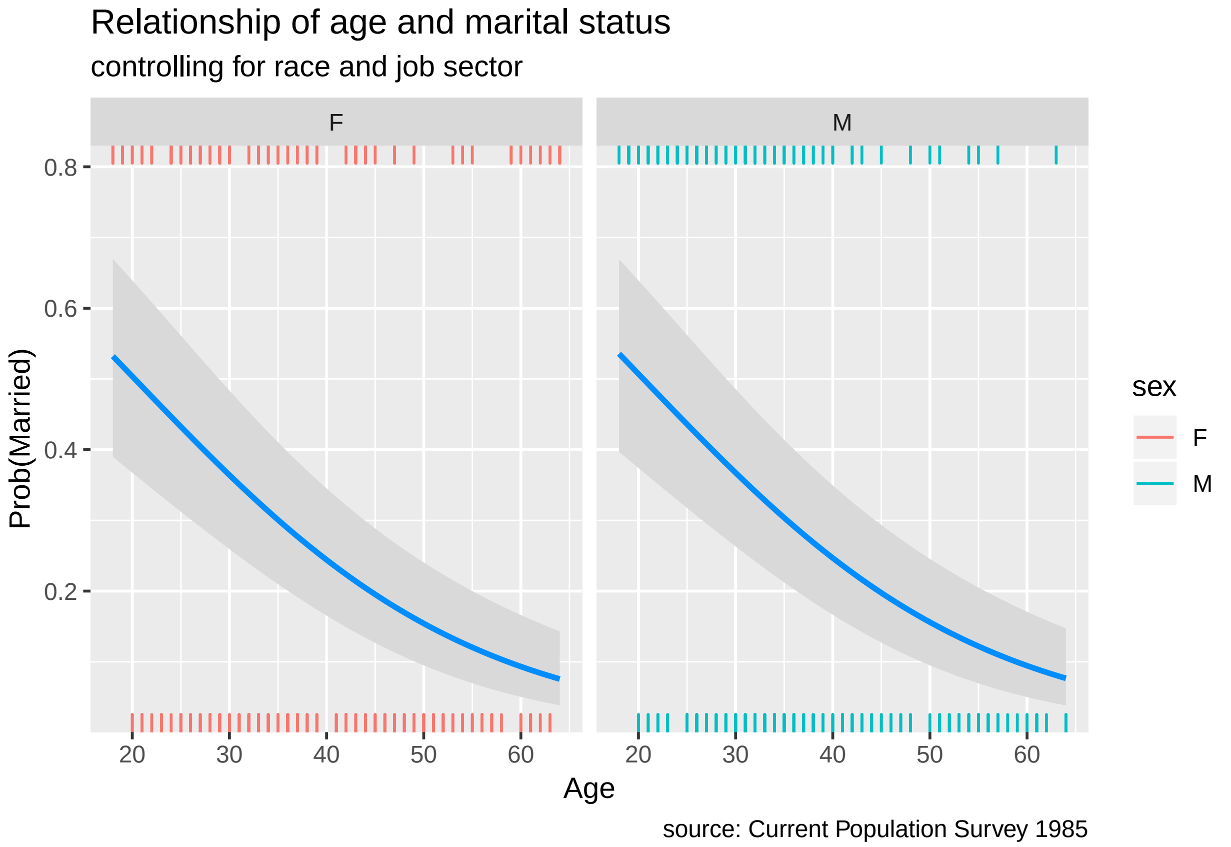
```r
# plot results
library(ggplot2)
library(visreg)
visreg(cps85_glm, "age",
       gg = TRUE,
       scale="response") +
  labs(y = "Prob(Married)",
       x = "Age",
       title = "Relationship of age and marital status",
       subtitle = "controlling for sex, race, and job sector",
       caption = "source: Current Population Survey 1985")
```



The probability of being married is estimated to be roughly 0.5 at age 20 and decreases to 0.1 at age 60, controlling for the other variables.

We can create multiple conditional plots by adding a `by` option. For example, the following code will plot the probability of being married by age, seperately for men and women, controlling for race and job sector.

```
# plot results
library(ggplot2)
library(visreg)
visreg(cps85_glm, "age",
       by = "sex",
       gg = TRUE,
       scale="response") +
  labs(y = "Prob(Married)",
       x = "Age",
       title = "Relationship of age and marital status",
       subtitle = "controlling for race and job sector",
       caption = "source: Current Population Survey 1985")
```



In this data, the probability of marriage is very similar for men and women.

8.4 Survival plots

In many research settings, the response variable is the time to an event. This is frequently true in healthcare research, where we are interested in time to recovery, time to death, or time to relapse.

If the event has not occurred for an observation (either because the study ended or the patient dropped out) the observation is said to be *censored*.

The `NCCTG Lung Cancer` dataset in the `survival` package provides data on the survival times of patients with advanced lung cancer following treatment. The study followed patients for up 34 months.

The outcome for each patient is measured by two variables

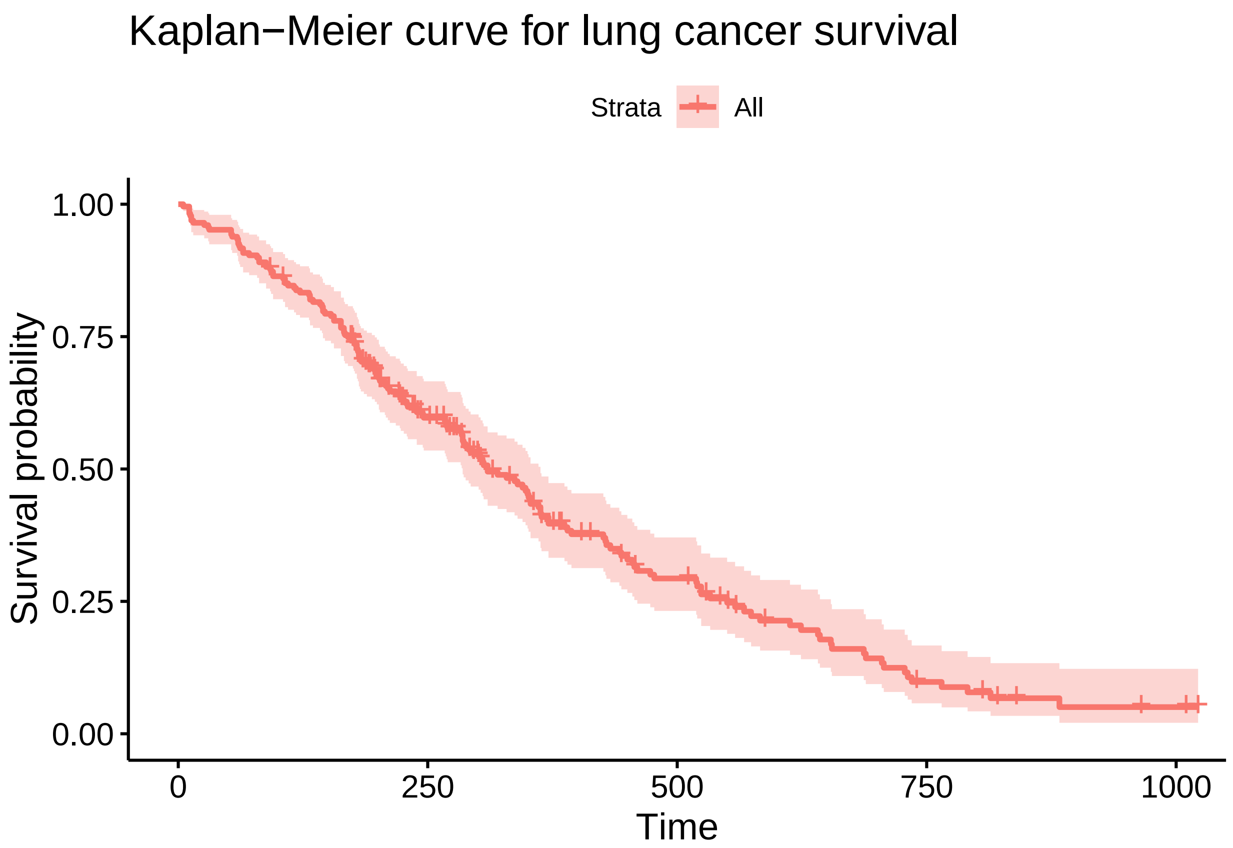
- *time* - survival time in days
- *status* - 1=censored, 2=dead

Thus a patient with *time=305 & status=2* lived 305 days following treatment. Another patient with *time=400 & status=1*, lived **at least** 400 days but was then lost to the study. A patient with *time=1022 & status=1*, survived to the end of the study (34 months).

A survival plot (also called a Kaplan-Meier Curve) can be used to illustrates the probability that an individual survives up to and including time *t*.

```
# plot survival curve
library(survival)
library(survminer)

data(lung)
sfit <- survfit(Surv(time, status) ~  1, data=lung)
ggsurvplot(sfit,
            title="Kaplan-Meier curve for lung cancer survival")
```



Roughly 50% of patients are still alive 300 days post treatment. Run `summary(sfit)` for more details.

It is frequently of great interest whether groups of patients have the same survival probabilities. In the next graph, the survival curve for men and women are compared.

```
# plot survival curve for men and women
sfit <- survfit(Surv(time, status) ~  sex, data=lung)
ggsurvplot(sfit,
           conf.int=TRUE,
           pval=TRUE,
           legend.labs=c("Male", "Female"),
           legend.title="Sex",
           palette=c("cornflowerblue", "indianred3"),
           title="Kaplan-Meier Curve for lung cancer survival",
           xlab = "Time (days)")
```

The `ggsurvplot` has many options. In particular, `conf.int` provides confidence intervals, while `pval` provides a log-rank test comparing the survival curves.

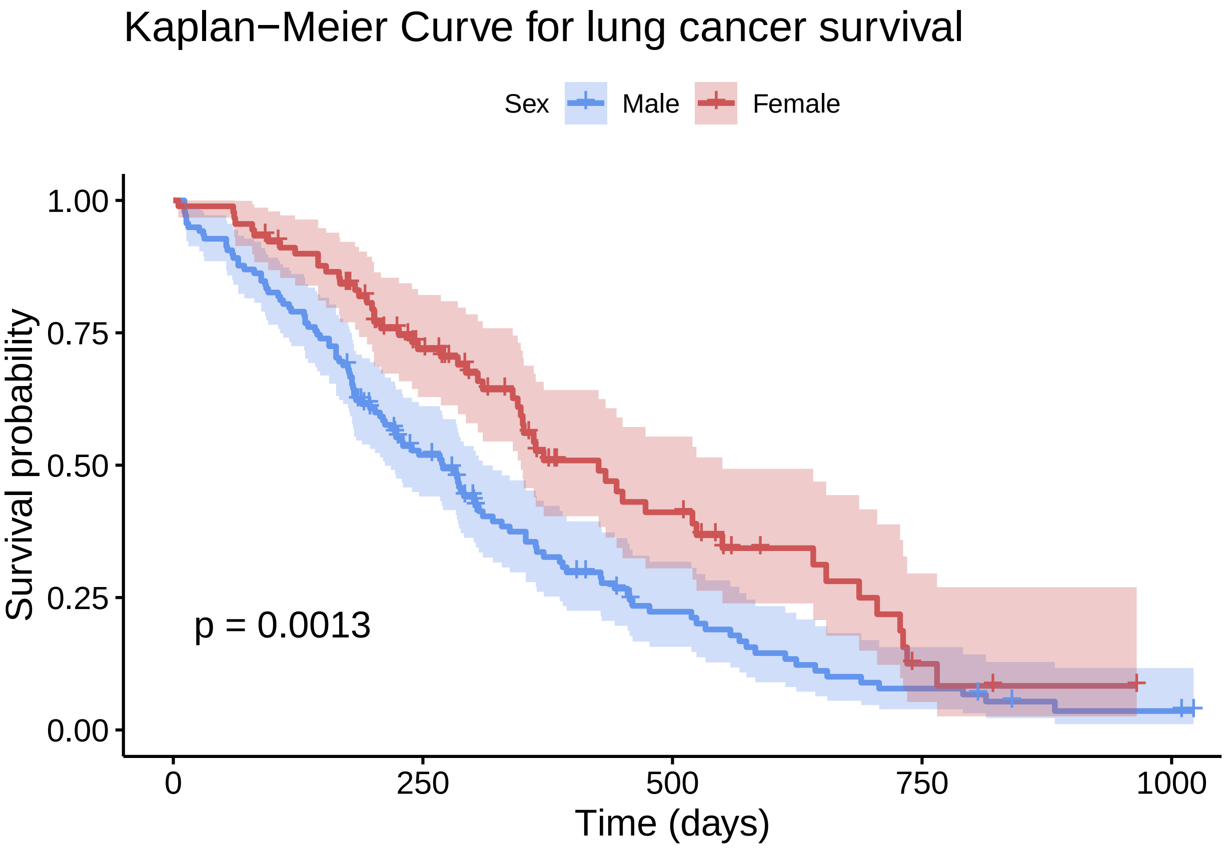


Figure 8.5: Comparison of survival curve

The p-value (0.0013) provides strong evidence that men and women have different survival probabilities following treatment.

8.5 Mosaic plots

Mosaic charts can display the relationship between categorical variables using rectangles whose areas represent the proportion of cases for any given combination of levels. The color of the tiles can also indicate the degree relationship among the variables.

Although mosaic charts can be created with `ggplot2` using the `ggmosaic` package, I recommend using the `vcd` package instead. Although it won't create `ggplot2` graphs, the package provides a more comprehensive approach to visualizing categorical data.

People are fascinated with the Titanic (or is it with Leo?). In the Titanic disaster, what role did sex and class play in survival? We can visualize the relationship between these three categorical variables using the code below.

```
# input data
library(readr)
titanic <- read_csv("titanic.csv")

# create a table
tbl <- xtabs(~Survived + Class + Sex, titanic)
ftable(tbl)
```

```
##                 Sex Female Male
## Survived Class
## No       1st            4  118
##          2nd           13  154
##          3rd          106  422
##          Crew           3  670
## Yes      1st          141   62
##          2nd           93   25
##          3rd           90   88
##          Crew          20  192
```

```
# create a mosaic plot from the table
library(vcd)
mosaic(tbl, main = "Titanic data")
```

The size of the tile is proportional to the percentage of cases in that combination of levels. Clearly more passengers perished, than survived. Those that perished were primarily 3rd class male passengers and male crew (the largest group).

If we assume that these three variables are independent, we can examine the residuals from the model and shade the tiles to match. In the graph below, dark blue represents more cases than expected given independence. Dark red represents less cases than expected if independence holds.

```
mosaic(tbl,
       shade = TRUE,
       legend = TRUE,
       labeling_args = list(set_varnames = c(Sex = "Gender",
                                             Survived = "Survived",
```

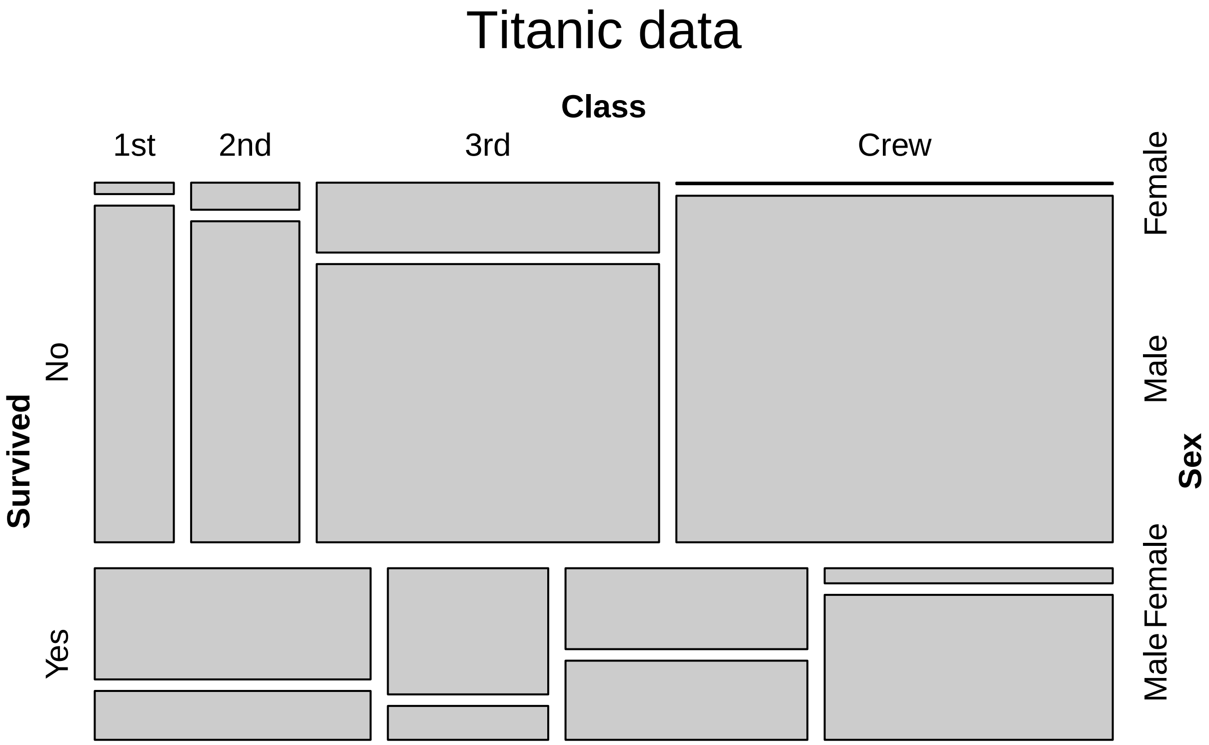


Figure 8.6: Basic mosaic plot

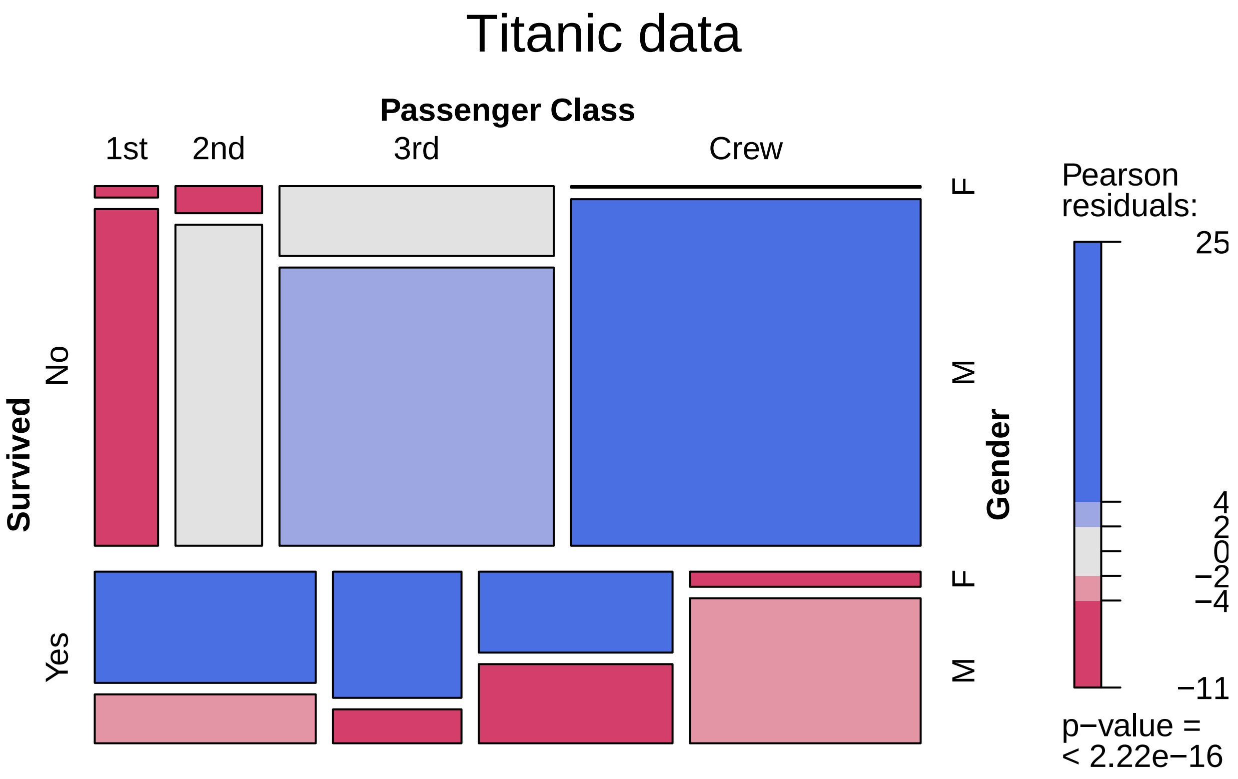


Figure 8.7: Mosaic plot with shading

```
                                          Class = "Passenger Class")),
        set_labels = list(Survived = c("No", "Yes"),
                          Class = c("1st", "2nd", "3rd", "Crew"),
                          Sex = c("F", "M")),
        main = "Titanic data")
```

We can see that if class, gender, and survival are independent, we are seeing many more male crew perishing, and 1st, 2nd and 3rd class females surviving than would be expected. Conversely, far fewer 1st class passengers (both male and female) died than would be expected by chance. Thus the assumption of independence is rejected. (Spoiler alert: Leo doesn't make it.)

For complicated tables, labels can easily overlap. See `labeling_border`, for plotting options.