An Introduction to R

Theresa Scott, MS

August 16, 2004

Contents

1 What Is R? Why Use R?

According to the R website (`http://www.R-project.org`):

- Language and environment for statistical computing and graphics
- GNU project, similar to the S Language (S-Plus), and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues
- Available as *free* software
- Runs on several UNIX platforms, Linux platforms, Windows, and MacOS
- *Greatest strength:* Ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed
- Open-source and highly extensible; constantly expanding
  - Can be extended beyond basic statistics via *packages*
- Able to define new *functions*, increasing its functionality, and run extensive simulations ("looping")
- Because it is command driven, analysis and results are completely *reproducible* if properly documented
  - Not always true with menu driven analysis packages; much harder to document

2 Sourcing / Downloading R

- R Website: `http://www.R-project.org`
- Windows (95 and later), Linux/UNIX, MacOS X
- To download:
  - Under Download link: CRAN > United States of America
  - `http://cran.r-project.org/`

3 Helpful References

- Documentation section of R website
  - Manuals; FAQs; contributed manuals, tutorials, etc. by users of R (e.g. Frank Harrell); newsletter, help pages, publications
- Biostats department website: `http://biostat.mc.vanderbilt.edu`
  - Statistical Computing > R and S-Plus Software and Documentation
- Other publications:
  - *Simple R - Using R for Introductory Statistics* by John Verzani (pdf)
  - *An Introduction to S and the Hmisc and Design Libraries* by Carlos Alzola and Frank E. Harrell (pdf)
  - *Data Analysis and Graphics Using R* by John Maindonald and John Braun (book)
  - *Statistical Tables and Plots Using S and LaTex* by Frank E. Harrell (pdf)
  - *R Data Import/Export* by the R Development Core Team (pdf)
  - *R for Beginners* by Emmanuel Paradis (pdf)
  - *An Introduction to R* by W.N. Venables, D.M. Smith, and the R Development Core Team (pdf or book)
  - *Introductory Statistics with R* by Peter Dalgaard (book)
  - *Regression Modelling Strategies* by Frank E. Harrell (book)
  - Not R specific, but good sources:
    - *The Elements of Graphing Data* and *Visualizing Data* by William S. Cleveland (books)

4 Interacting with R

- R evaluates and prints out the result of any *expression* that one enters in at the *command line prompt* in the *console window*[1]
- The result, if any, appears on subsequent lines
- **Simplest use of R:** Using R as a calulator

```
> 2 + 2

[1] 4

> sqrt(10)

[1] 3.162278
```

- **Most frequent use of R:** Using R to evaluate *expressions*, which include *functions* and defined *objects*

5 R Objects

- All R entities, including *functions* and *data structures*, exist as *objects*[2]
- They can all be operated on as data within *expressions*
- If you type the name of an *object* at the command prompt, the contents of the *object* are printed out (e.g. type `q`, `mean`)
  - *Objects* are case sensitive (e.g. `Age` and `age` would refer to two different *objects*)
- **Managing Project Data in R:**
  - By default, R stores all the objects created in your session in a single file: `.RData`, which is directory specific
  - When running R interactively, R asks whether you want to update `.RData` to contain newly created objects upon termination of the session
  - Since many of the objects are temporary, best to answer `n` ("No") to this question and not use the `.RData` mechanism
  - Instead, use the `save` function to store some of your newly created data frames and selected other objects (i.e. regression fit objects that took significant execution time to create) permanently

[1] *Data Analysis Using R*, Maindonald
[2] *Data Analysis Using R*, Maindonald

- Use the `save`'s `compress` argument to store the resulting file very compactly
- *Example*: A hypothetical data set containing a sample of 500 subjects. Each had three potential predictors measured: age, sex, and systolic blood pressure, and a diagnosis of a certain disease (present/absent).

```
> library(Hmisc)
> library(Design)
> prob1 <- read.table("prob1.csv", header = T, sep = ",")
> prob1 <- upData(prob1, labels = c(age = "Age", sex = "Sex", sysbp = "Systolic Blo
+     dz = "Disease"), units = c(age = "years"), levels = list(sex = c("Female",
+     "Male")))
Input object size:         18688 bytes;         4 variables
New object size:         19544 bytes;         4 variables
> m1 <- lm(age ~ sysbp + sex, data = prob1)
> save(m1, prob1, file = "prob1.rda", compress = TRUE)
```

- To retrieve the saved objects in a future session, use the `load` function:

```
> load("prob1.rda")
```

6 Functions

- Almost everything in R is done by calling *functions*
- Most functions have *arguments* that pass values to the function for it to work on or to specify detailed options on how it should do its work[3]
- *Arguments* are given to the function either by name or by their sequential position in the series of arguments
- The 12,000 (and growing) functions in R are organized into *packages*, some of which are loaded when you start R, while others must be loaded explicitly using the `library` function
  - The *Hmisc* package (i.e. "Harrell Miscellaneous"), which was developed by Frank E. Harrell, contains many functions useful for data analysis, high-level graphics, utility operations, functions for computing sample size and power, importing datasets, imputing missing values, advanced table making, variable clustering, character string manipulation, conversion of S objects to LaTeX code, and recoding variables.[4]
  - The *Design* library, which was also developed by Frank E. Harrell, is a collection of about 180 functions that assist and streamline regression modeling, testing, estimation, validation, graphics, prediction, and typesetting by storing enhanced model design attributes in the fit.[5]

[3] *An Introduction to S and the Hmisc and Design Libraries*, Harell
[4] *An Introduction to S and the Hmisc and Design Libraries*, Harell
[5] The *Information on Package 'Design'* help page in R (`library(help="Design")`)

7 R Details

- **Command line prompt:** >
  - R commands (*expressions*) are typed following this prompt
  - *Example:*

    ```
    > mean(c(12, 10, 20, 15, 30))
    [1] 17.4
    ```

- **Continuation prompt:** +
  - Appears in command window when, following a carriage return, the command is still not complete
  - *Example:*

    ```
    > mean(c(5, 4, 3, 7, 9, NA, 10, 19, 18, 14, 2, 15, 22, 35, 49,
    +     NA, NA, 54), na.rm = T)
    [1] 17.73333
    ```

- **Comment:** #
  - Anything following a # on the command line is taken as "comment" and ignored by R
  - *Example:* `> 175*(8/5) # convert 175 miles to kms`

- **Multiple command separator:** ;
  - Allows multiple commands to appear on one line
  - *Example:* `> exp(1); 200-15`

- **Assigmnent:** <-
  - Left diamond bracket (<) followed by a minus sign (-)
  - *Example:* x <- 2 implies "x is assigned to 2"

    ```
    > x <- 2
    > x
    [1] 2
    ```

- **Arithmetic operators:** +, -, *, /, ^ , exp, log, log10
- **Comparison operators:** <, >, <=, >=, ==, !=
- **Logical operators:** &, |, !

8 Importing Data

- The easiest form of data import into R is a simple text file[6]
  - Often have the data saved as an Excel spreadsheet, SPSS file, or STATA file
  - Export the data as a text file (e.g. a delimited file; either tab-delimited or comma-delimited)
- ***General Tip:*** Start R session from within the relevant project *directory* (folder)
  - R automatically stores all *objects* created in your R session to your *workspace* (.Rdata file), which is directory specific
  - Allows you to keep defined *objects* separate for each project and not overload your *workspace*
  - Allows you to easily reference your data file for importing
- *Example:* the *Low Birthweight Study data* (Hosmer & Lemeshow, *Applied Logistic Regression*) read in as a tab-delimited file
  - The original file is an Excel spreadsheet (.xls)
  - *NOTE:* No missing values, so do not need to worry about specifying them
  - Use Excel to save the file as a tab-delimited file (.txt)
    - File > Save As: Save As Type: Text (Tab delimited) (*.txt)
  - In R:

```
> lowbw <- read.table("lowbwt.txt", header = T)
```

- *Example:* the *Primary Biliary Cirrhosis (PBC) Trial data* (Fleming & Harrington, *Counting Processes and Survival Analysis*) read in as a comma-delimited file
  - The original file is an Excel spreadsheet (.xls)
  - Use Excel to save the file as a comma-delimited file (.csv)
    - File > Save As: Save As Type: CSV (Comma delimited) (*.csv)
  - *NOTE:* Data contains missing values:
    - In Excel, replace all blank cells with "NA" (before saving as .csv)
      - Edit > Find & Replace: Search for: (blank), Replace with "NA"
    - In R (after saving as .csv), specify `na.string = ""` in `read.table` function

[6] *An Introduction to S and the Hmisc and Design Libraries*, Harell; *R Data Import/Export*, R Development Core Team

- In R:

```
> pbc <- read.table("liver.csv", header = T, na.string = "", sep = ",")
```

- For STATA files:

  1. To read in a STATA file (.dta) directly, use the `read.dta` function in the *foreign* library (example given in section 12.1)
  2. To create an ASCII file from STATA, enter the following commands in STATA (using `estriol.dta` for illustration):
     - Space-delimited text file: `outfile using estriol.dta wide`
     - Comma-delimited text file: `outfile using estriol.dta wide comma`
     - *NOTE:* The option `wide` ensures one record per line; otherwise the records are wrapped to the next line

- ***General Tip:*** Always ensure your data set has been read-in (imported) correctly

  - Use the `dim` function to check the dimensions (number of rows, number of columns) of your read-in *data frame*

```
> dim(lowbw)
```

```
[1] 189  11
```

```
> dim(pbc)
```

```
[1] 310  20
```

  - Use the `names` function to check the names of the columns of your read-in *data frame*

```
> names(lowbw)
```

```
 [1] "id"    "low"   "age"   "lwt"   "race"  "smoke" "ptl"   "ht"    "ui"
[10] "ftv"   "bwt"
```

```
> names(pbc)
```

```
 [1] "age"      "albumin"  "alkphos"  "ascites"  "bili"     "cholest"
 [7] "edema"    "edmadj"   "hepmeg"   "obstime"  "platelet" "protime"
[13] "sex"      "sgot"     "spiders"  "stage"    "status"   "tx"
[19] "trig"     "urinecu"
```

  - Use the *Hmisc* `contents` function to check the following attributes of the variables from your read-in *data frame*: names, labels (if any), units (if any), number of factor levels (if any), factor levels, class, storage mode, and number of NAs

```
> library(Hmisc)
```

```
> contents(lowbw)
```

```
Data frame:lowbw        189 observations and 11 variables    Maximum # NAs:0

      Storage
id    integer
low   integer
age   integer
lwt   integer
race  integer
smoke integer
ptl   integer
ht    integer
ui    integer
ftv   integer
bwt   integer
```

> contents(pbc)

```
Data frame:pbc        310 observations and 20 variables    Maximum # NAs:30

         Storage NAs
age       double   0
albumin   double   0
alkphos   double   0
ascites  integer   0
bili      double   0
cholest  integer  28
edema    integer   0
edmadj    double   0
hepmeg   integer   0
obstime  integer   0
platelet integer   4
protime   double   0
sex      integer   0
sgot      double   0
spiders  integer   0
stage    integer   0
status   integer   0
tx       integer   0
trig     integer  30
urinecu  integer   2
```

- ***General Tip:*** Make any changes to the read-in data frame (e.g. variable names, labels, or value codes) upfront in order to take advantage of all the new annotations during your analysis

  – Use the *Hmisc* `upData` function to edit the read-in data frame's *contents* (e.g. variable names, labels, levels, units, etc.)

– The *Hmisc* `upData` function accomplishes the following, listed in order in which changes are executed by the function:

1. optionally change names of variables to lower case
2. rename variables
3. adds new variables
4. recomputes existing variables from the original variable and/or from other variables in the data frame
5. changes the storage mode of variables to the most efficient mode (as done with `clean.import`)
6. drops variables
7. adds, changes, and combines `levels` of `factor` variables
8. adds or changes variable `labels` attributes
9. adds or changes variable `units` (units of measurement) attributes

```
> library(Hmisc)
> lowbw <- upData(lowbw, labels = c(id = "Subject Identification Code",
+     low = "Low Birthweight?", age = "Mother's Age", lwt = "Mother's Weight at Last Menstru
+     race = "Race", smoke = "Did Mother Smoke During Pregnancy?",
+     ptl = "Number of Premature Labors", ht = "History of Hypertension?",
+     ui = "Uterine Irritability?", ftv = "Number of Physician Visits in 1st Trimester",
+     bwt = "Birthweight"), units = c(age = "years", lwt = "lbs",
+     bwt = "grams"), levels = list(low = c(">2500g", "<=2500g"),
+     race = c("White", "Black", "Other"), smoke = c("No", "Yes"),
+     ht = c("No", "Yes"), ui = c("No", "Yes")))

Input object size:         11428 bytes;         11 variables
New object size:          14844 bytes;         11 variables

> contents(lowbw)

Data frame:lowbw        189 observations and 11 variables    Maximum # NAs:0

                                          Labels Units Levels Storage
id                    Subject Identification Code              integer
low                               Low Birthweight?            2 integer
age                                   Mother's Age years        integer
lwt      Mother's Weight at Last Menstrual Period   lbs        integer
race                                        Race              3 integer
smoke          Did Mother Smoke During Pregnancy?             2 integer
ptl                    Number of Premature Labors              integer
ht                       History of Hypertension?            2 integer
ui                          Uterine Irritability?            2 integer
ftv   Number of Physician Visits in 1st Trimester              integer
bwt                                  Birthweight grams        integer
```

```
+--------+-----------------+
|Variable|Levels           |
+--------+-----------------+
|  low   |>2500g,<=2500g   |
+--------+-----------------+
|  race  |White,Black,Other|
+--------+-----------------+
|  smoke |No,Yes           |
+--------+-----------------+
|  ht    |No,Yes           |
+--------+-----------------+
|  ui    |No,Yes           |
+--------+-----------------+
```

```
> pbc <- upData(pbc, labels = c(age = "Age", albumin = "Serum Albumin",
+     alkphos = "Serum Alkaline Phosphatase", ascites = "Presense of Ascites",
+     bili = "Serum Bilirubin", cholest = "Serum Cholesterol",
+     edema = "Presence of Edema", edmadj = "Graded Measurement of Edema",
+     hepmeg = "Presence of Hepatomegaly", obstime = "Observation Time",
+     platelet = "Platelet Count", protime = "Prothrombin Time",
+     sex = "Sex", sgot = "Serum SGOT", spiders = "Presence of Spider Angiomata",
+     stage = "Stage of Disease", status = "Survival Status", tx = "Treatment Group",
+     trig = "Serum Triglycerides", urinecu = "Urine Copper"),
+     units = c(age = "year", obstime = "day"), levels = list(ascites = c("Absent",
+         "Present"), edema = c("Absent", "Present"), hepmeg = c("Absent",
+         "Present"), sex = c("Male", "Female"), spiders = c("Absent",
+         "Present"), stage = c("Best", "Better", "Worse", "Worst"),
+         status = c("Censored", "Died"), tx = c("Placebo", "Drug")))

Input object size:         38392 bytes;         20 variables
New object size:          43976 bytes;         20 variables

> contents(pbc)

Data frame:pbc        310 observations and 20 variables    Maximum # NAs:30

                                Labels Units Levels Storage NAs
age                                Age  year         double   0
albumin                  Serum Albumin               double   0
alkphos    Serum Alkaline Phosphatase               double   0
ascites           Presense of Ascites            2 integer   0
bili                  Serum Bilirubin               double   0
cholest             Serum Cholesterol              integer  28
edema               Presence of Edema            2 integer   0
edmadj    Graded Measurement of Edema               double   0
hepmeg       Presence of Hepatomegaly            2 integer   0
obstime              Observation Time   day        integer   0
```

```
platelet               Platelet Count              integer   4
protime              Prothrombin Time               double   0
sex                               Sex            2 integer   0
sgot                        Serum SGOT               double   0
spiders  Presence of Spider Angiomata            2 integer   0
stage                Stage of Disease            4 integer   0
status                Survival Status            2 integer   0
tx                    Treatment Group            2 integer   0
trig               Serum Triglycerides              integer  30
urinecu                   Urine Copper              integer   2
```

Variable	Levels
ascites	Absent,Present
edema	Absent,Present
hepmeg	Absent,Present
sex	Male,Female
spiders	Absent,Present
stage	Best,Better,Worse,Worst
status	Censored,Died
tx	Placebo,Drug

9 Data Summaries

There are many functions to produce statistical summaries (including the `mean`, `median`, `sd`, and `table` functions), but here is an illustration of a few of the more "advanced" ones in the *Hmisc* library:[7]

9.1 The `table` function

```
> table(pbc$tx)

Placebo    Drug
    153     157

> table(pbc$tx, pbc$status)
```

[7] *An Introduction to S and the Hmisc and Design Libraries*, Harell

```
          Censored Died
  Placebo 93       60
  Drug    92       65
```

```
> with(pbc, table(tx, status))
```

```
         status
tx        Censored Died
  Placebo 93       60
  Drug    92       65
```

9.2 The *Hmisc* `describe` function

```
> library(Hmisc)
> describe(lowbw)
```

```
lowbw

 11  Variables      189  Observations
---------------------------------------------------------------------------
id : Subject Identification Code
      n missing  unique    Mean     .05     .10     .25     .50     .75     .90
    189       0     189   121.1    20.8    30.8    68.0   123.0   176.0   207.2
    .95
  216.6

lowest :  4  10  11  13  15 , highest: 222 223 224 225 226
---------------------------------------------------------------------------
low : Low Birthweight?
      n missing  unique
    189       0       2

>2500g (130, 69%), <=2500g (59, 31%)
---------------------------------------------------------------------------
age : Mother's Age [years]
      n missing  unique    Mean     .05     .10     .25     .50     .75     .90
    189       0      24   23.24      16      17      19      23      26      31
    .95
     32

lowest : 14 15 16 17 18, highest: 33 34 35 36 45
---------------------------------------------------------------------------
lwt : Mother's Weight at Last Menstrual Period [lbs]
      n missing  unique    Mean     .05     .10     .25     .50     .75     .90
    189       0      75   129.8    94.4    99.6   110.0   121.0   140.0   170.0
    .95
  188.2
```

```
lowest : 80  85  89  90  91 , highest: 215 229 235 241 250
---------------------------------------------------------------------------
race : Race
      n missing  unique
    189       0       3

White (96, 51%), Black (26, 14%), Other (67, 35%)
---------------------------------------------------------------------------
smoke : Did Mother Smoke During Pregnancy?
      n missing  unique
    189       0       2

No (115, 61%), Yes (74, 39%)
---------------------------------------------------------------------------
ptl : Number of Premature Labors
      n missing  unique    Mean
    189       0       4  0.1958

0 (159, 84%), 1 (24, 13%), 2 (5, 3%), 3 (1, 1%)
---------------------------------------------------------------------------
ht : History of Hypertension?
      n missing  unique
    189       0       2

No (177, 94%), Yes (12, 6%)
---------------------------------------------------------------------------
ui : Uterine Irritability?
      n missing  unique
    189       0       2

No (161, 85%), Yes (28, 15%)
---------------------------------------------------------------------------
ftv : Number of Physician Visits in 1st Trimester
      n missing  unique    Mean
    189       0       6  0.7937

            0  1  2 3 4 6
Frequency 100 47 30 7 4 1
%          53 25 16 4 2 1
---------------------------------------------------------------------------
bwt : Birthweight [grams]
      n missing  unique    Mean     .05     .10     .25     .50     .75     .90
    189       0     133    2945    1801    2038    2414    2977    3475    3865
    .95
   3997
```

```
lowest :  709 1021 1135 1330 1474, highest: 4167 4174 4238 4593 4990
---------------------------------------------------------------------------
```

```
> describe(pbc$edmadj)
```

```
pbc$edmadj : Graded Measurement of Edema
      n missing  unique    Mean
    310       0       3  0.1113

0.0 (261, 84%), 0.5 (29, 9%), 1.0 (20, 6%)
```

```
> describe(pbc[, c("bili", "stage")])
```

```
pbc[, c("bili", "stage")]

 2  Variables      310  Observations
---------------------------------------------------------------------------
bili : Serum Bilirubin
      n missing  unique    Mean     .05     .10     .25     .50     .75     .90
    310       0      84    3.27   0.500   0.600   0.800   1.350   3.475   7.210
    .95
 14.055

lowest : 0.3  0.4  0.5  0.6  0.7 , highest: 21.6 22.5 24.5 25.5 28.0
---------------------------------------------------------------------------
stage : Stage of Disease
      n missing  unique
    310       0       4

Best (16, 5%), Better (65, 21%), Worse (120, 39%), Worst (109, 35%)
---------------------------------------------------------------------------
```

9.3 The *Hmisc* bystats function

```
> library(Hmisc)
> library(Design)
> bystats(pbc$age, pbc$status)

 Mean of pbc$age by

           N     Mean
Censored 185 47.73017
Died     125 53.24174
ALL      310 49.95257
```

```
> summary(age ~ status, data = pbc)
```

```
Age    N=310

+---------------+--------+---+--------+
|               |        |N  |age     |
+---------------+--------+---+--------+
|Survival Status|Censored|185|47.73017|
|               |Died    |125|53.24174|
+---------------+--------+---+--------+
|Overall        |        |310|49.95257|
+---------------+--------+---+--------+
```

```
> bystats(pbc$albumin, pbc$tx, pbc$status, fun = quantile)

 quantile of pbc$albumin by pbc$tx, pbc$status

                   N   0%  25%   50%   75% 100%
Placebo Censored  93 2.90 3.41 3.610 3.830 4.38
Drug    Censored  92 2.83 3.40 3.665 3.855 4.64
Placebo Died      60 1.96 3.19 3.435 3.670 4.30
Drug    Died      65 2.10 3.05 3.350 3.700 4.40
ALL              310 1.96 3.31 3.555 3.800 4.64
```

```
> summary(albumin ~ tx + status, method = "cross", data = pbc,
+     fun = quantile)

 UseMethod by tx, status

+----+
|N   |
|0%  |
|25% |
|50% |
|75% |
|100%|
+----+
+-------+--------+-----+-----+
|   tx  |Censored| Died| ALL |
+-------+--------+-----+-----+
|Placebo|   93   | 60  |153  |
|       |  2.90  |1.96 |1.96 |
|       |  3.41  |3.19 |3.35 |
|       |  3.610 |3.435|3.550|
|       |  3.830 |3.670|3.780|
|       |  4.38  |4.30 |4.38 |
+-------+--------+-----+-----+
|Drug   |   92   | 65  |157  |
|       |  2.83  |2.10 |2.10 |
```

```
|       |  3.40  |3.05 |3.21 |
|       |  3.665 |3.350|3.570|
|       |  3.855 |3.700|3.830|
|       |  4.64  |4.40 |4.64 |
+-------+--------+-----+-----+
|ALL    |  185   |125  |310  |
|       |  2.83  |1.96 |1.96 |
|       |  3.40  |3.11 |3.31 |
|       |  3.630 |3.430|3.555|
|       |  3.850 |3.670|3.800|
|       |  4.64  |4.40 |4.64 |
+-------+--------+-----+-----+
```

9.4 Summary Statistics

```
> library(Hmisc)
> library(Design)

> sublowbw <- lowbw[, -1]
> summ <- summary(~., data = sublowbw)
> latex(summ, size = "smaller", middle.bold = T, digits = 3, file = "")

> lowbw$low

Low Birthweight?
  [1] >2500g  <=2500g >2500g  >2500g  >2500g  >2500g  >2500g  >2500g  >2500g
 [10] <=2500g <=2500g >2500g  >2500g  >2500g  >2500g  <=2500g >2500g  >2500g
 [19] >2500g  >2500g  >2500g  >2500g  >2500g  >2500g  <=2500g <=2500g >2500g
 [28] >2500g  >2500g  <=2500g <=2500g >2500g  >2500g  <=2500g >2500g  <=2500g
 [37] >2500g  >2500g  <=2500g <=2500g >2500g  <=2500g >2500g  >2500g  >2500g
 [46] >2500g  >2500g  >2500g  >2500g  >2500g  >2500g  >2500g  >2500g  <=2500g
 [55] >2500g  >2500g  >2500g  <=2500g >2500g  >2500g  >2500g  <=2500g >2500g
 [64] <=2500g >2500g  >2500g  >2500g  >2500g  <=2500g <=2500g >2500g  >2500g
 [73] <=2500g >2500g  >2500g  >2500g  >2500g  >2500g  >2500g  >2500g  >2500g
 [82] >2500g  >2500g  >2500g  >2500g  >2500g  >2500g  <=2500g >2500g  >2500g
 [91] >2500g  >2500g  >2500g  >2500g  >2500g  >2500g  >2500g  <=2500g <=2500g
[100] <=2500g >2500g  <=2500g >2500g  >2500g  >2500g  >2500g  <=2500g >2500g
[109] >2500g  >2500g  <=2500g <=2500g >2500g  >2500g  >2500g  <=2500g >2500g
[118] <=2500g >2500g  >2500g  >2500g  >2500g  <=2500g >2500g  >2500g  <=2500g
[127] >2500g  >2500g  >2500g  <=2500g >2500g  <=2500g <=2500g >2500g  >2500g
[136] >2500g  >2500g  <=2500g >2500g  <=2500g >2500g  >2500g  >2500g  >2500g
[145] >2500g  >2500g  >2500g  <=2500g >2500g  >2500g  >2500g  >2500g  >2500g
[154] <=2500g <=2500g >2500g  <=2500g >2500g  >2500g  <=2500g <=2500g >2500g
[163] >2500g  >2500g  <=2500g >2500g  <=2500g <=2500g >2500g  >2500g  <=2500g
[172] >2500g  >2500g  <=2500g <=2500g <=2500g <=2500g <=2500g <=2500g >2500g
[181] >2500g  >2500g  <=2500g <=2500g >2500g  <=2500g <=2500g <=2500g <=2500g
Levels: >2500g <=2500g
```

Table 1: Descriptive Statistics ($N = 189$)

Low Birthweight? : ≤2500g		31% (59)
Mother's Age	years	19 **23** 26
Mother's Weight at Last Menstrual Period	lbs	110 **121** 140
Race : White		51% (96)
Black		14% (26)
Other		35% (67)
Did Mother Smoke During Pregnancy? : Yes		39% (74)
Number of Premature Labors : 0		84% (159)
1		13% (24)
2		3% (5)
3		1% (1)
History of Hypertension? : Yes		6% (12)
Uterine Irritability? : Yes		15% (28)
Number of Physician Visits in 1st Trimester : 0		53% (100)
1		25% (47)
2		16% (30)
3		4% (7)
4		2% (4)
6		1% (1)
Birthweight	grams	2414 **2977** 3475

a b c represent the lower quartile a, the median b, and the upper quartile c for continuous variables.

Numbers after percents are frequencies.

```
> unclass(lowbw$low)

  [1] 1 2 1 1 1 1 1 1 2 2 1 1 1 1 2 1 1 1 1 1 1 1 2 2 1 1 1 2 2 1 1 2 1 2 1
 [38] 1 2 2 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 1 2 2 1 1 2 1
 [75] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 2 2 1 2 1 1 1 1 2 1 1 1 2
[112] 2 1 1 1 2 1 2 1 1 1 1 2 1 1 2 1 1 1 2 1 2 2 1 1 1 1 2 1 2 1 1 1 1 1 1 1 2
[149] 1 1 1 1 1 1 2 2 1 2 1 1 2 2 1 1 1 2 1 2 2 1 1 2 1 1 2 2 2 2 2 2 1 1 1 2 2 1
[186] 2 2 2 2
attr(,"levels")
[1] ">2500g"  "<=2500g"
attr(,"label")
[1] "Low Birthweight?"

> unclass(lowbw$low) - 1

  [1] 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 0 0 1 0 1 0
 [38] 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 1 0 0 0 0 1 1 0 0 1 0
 [75] 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 0 0 1 0 0 0 1
[112] 1 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0 0 0 1 0 1 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1
[149] 0 0 0 0 0 1 1 0 1 0 0 1 1 0 0 0 1 0 1 1 0 0 1 0 0 1 1 1 1 1 1 0 0 0 1 1 0
[186] 1 1 1 1
attr(,"levels")
[1] ">2500g"  "<=2500g"
attr(,"label")
[1] "Low Birthweight?"

> sublowbw2 <- lowbw[, -c(1, 2)]
> low.summ <- summary(unclass(lowbw$low) - 1 ~ ., data = sublowbw2)
> latex(low.summ, middle.bold = T, file = "")
```

10 Graphics in R

As with the statstical summaries, there is a large variety of plotting functions in R. The following is an illustration of just a few:

10.1 Histograms

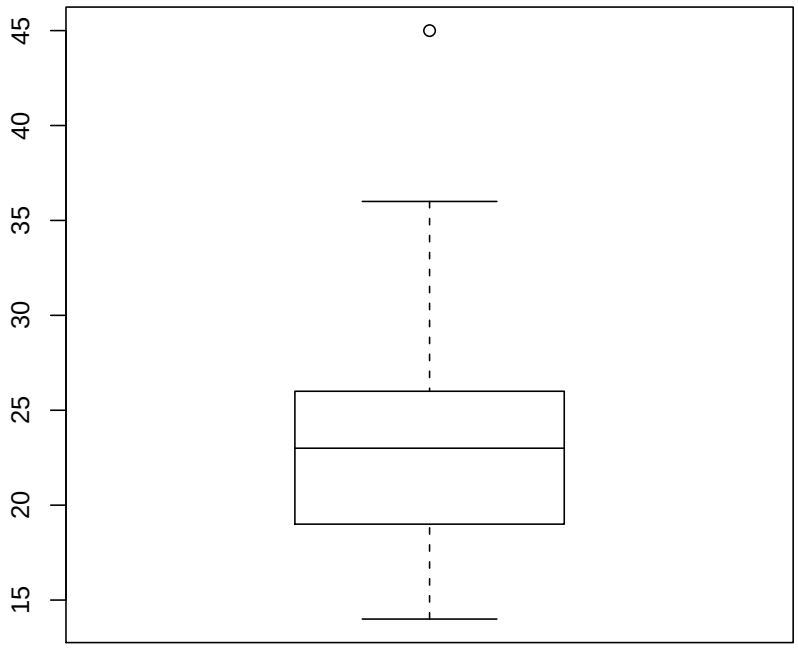
```
> hist(lowbw$age, main = "Histogram of Mother's Age", xlab = "Age (years)")
```

Table 2: Low Birthweight? N=189

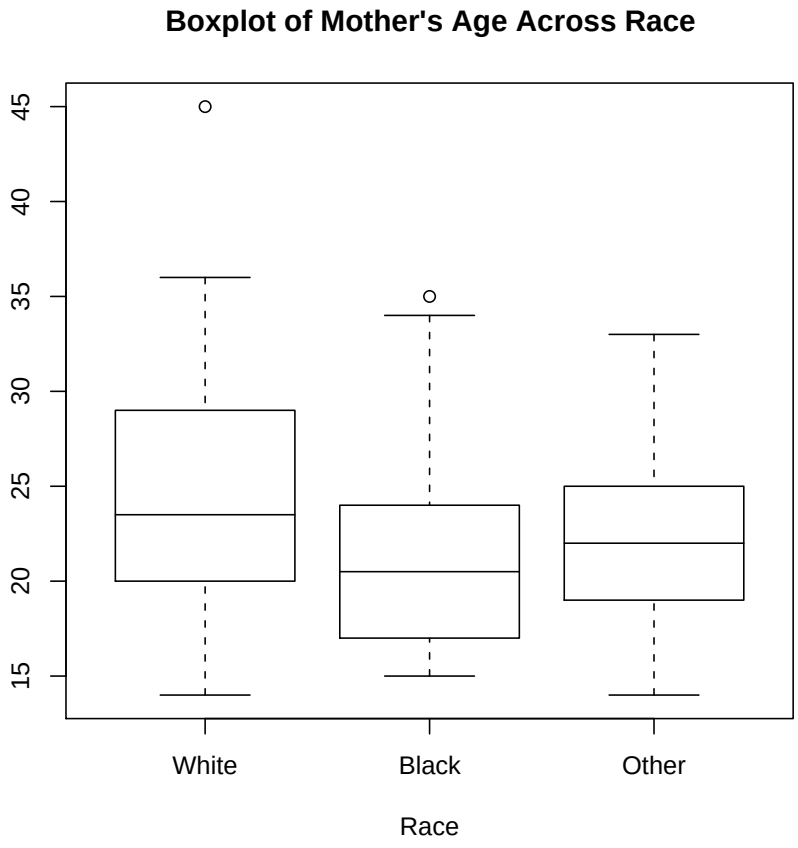
	N	unclass(lowbw$low) - 1
Mother's Age years		
[14,20)	51	0.29
[20,24)	56	0.36
[24,27)	36	0.42
[27,45]	46	0.20
Mother's Weight at Last Menstrual Period lbs		
[80,112)	53	0.47
[112,122)	43	0.23
[122,141)	46	0.26
[141,250]	47	0.26
Race		
White	96	0.24
Black	26	0.42
Other	67	0.37
Did Mother Smoke During Pregnancy?		
No	115	0.25
Yes	74	0.41
Number of Premature Labors		
0	159	0.26
1	24	0.67
2	5	0.40
3	1	0.00
History of Hypertension?		
No	177	0.29
Yes	12	0.58
Uterine Irritability?		
No	161	0.28
Yes	28	0.50
Number of Physician Visits in 1st Trimester		
0	100	0.36
1	47	0.23
2	30	0.23
3	7	0.57
4	4	0.25
6	1	0.00
Birthweight grams		
[709,2424)	48	1.00
[2424,2992)	48	0.23
[2992,3487)	46	0.00
[3487,4990]	47	0.00
Overall		
	189	0.31

10.2 Boxplots

```
> boxplot(lowbw$age, main = "Boxplot of Mother's Age", xlab = "Age (years)")
```

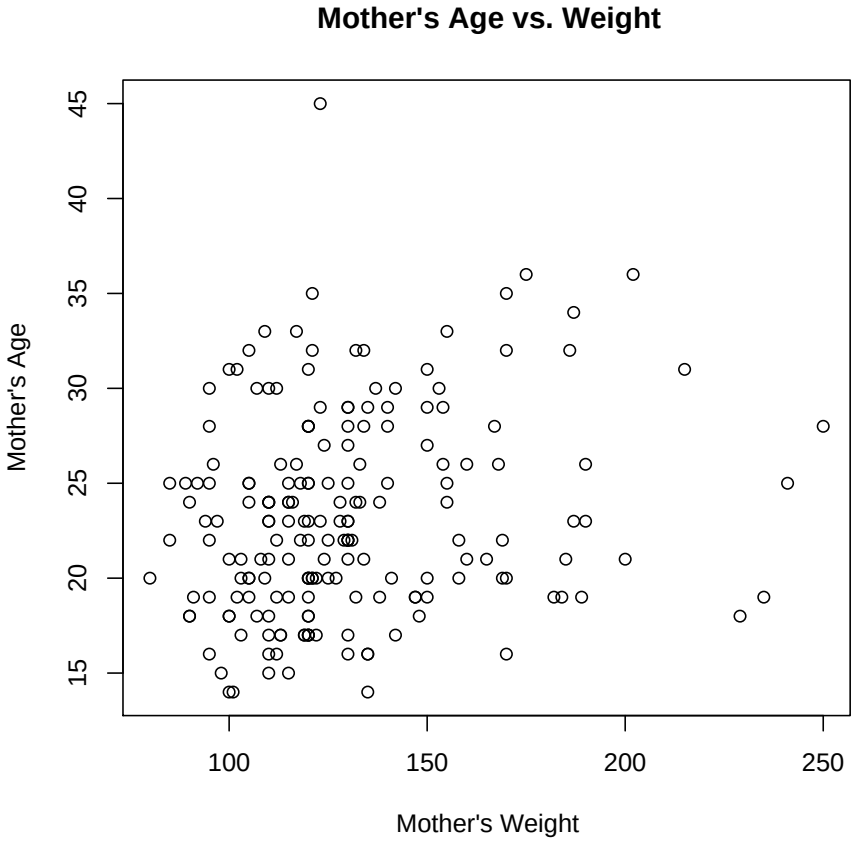



```
> boxplot(lowbw$age ~ lowbw$race, main = "Boxplot of Mother's Age Across Race",
+     xlab = "Race")
```

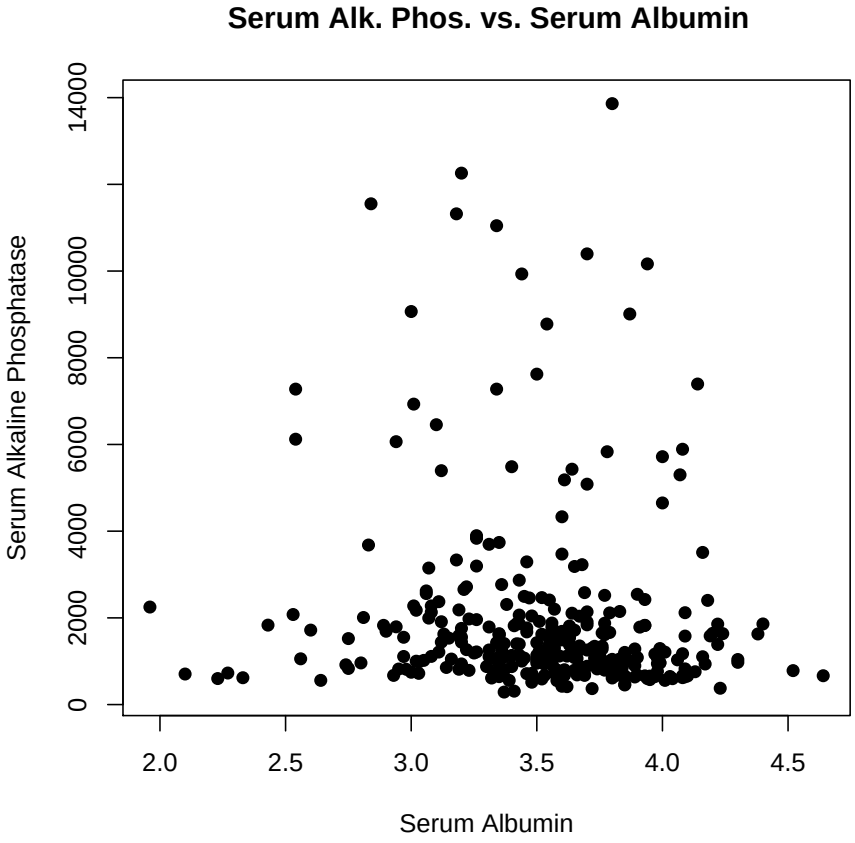


10.3 Scatter plots

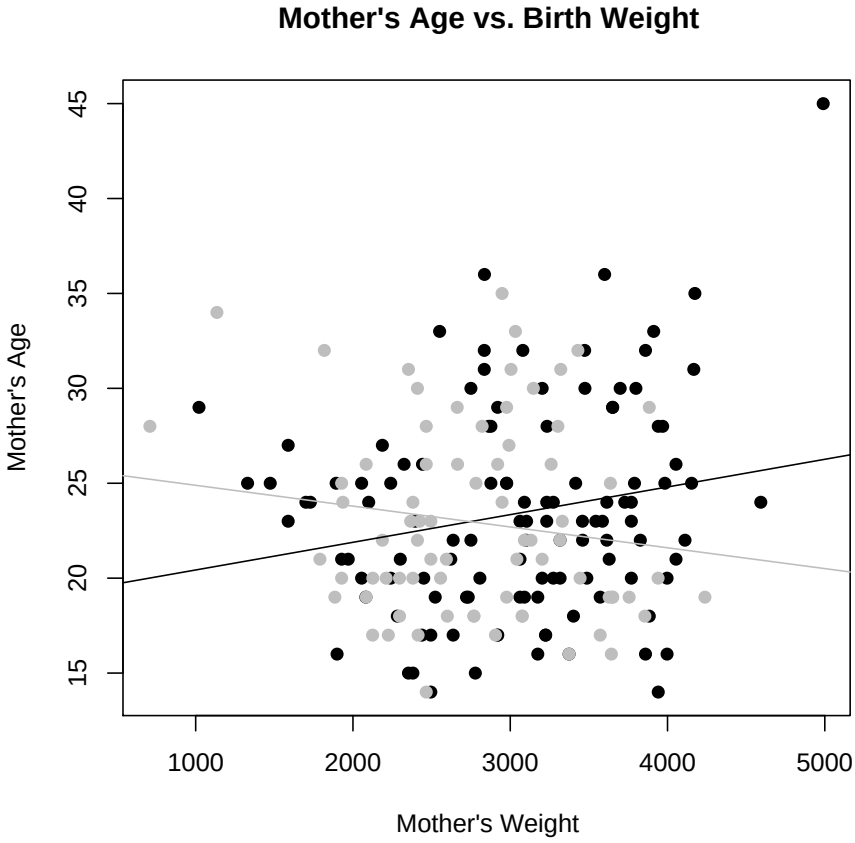
```
> plot(lowbw$age ~ lowbw$lwt, main = "Mother's Age vs. Weight",
+     xlab = "Mother's Weight", ylab = "Mother's Age")
```




```
> plot(pbc$albumin, pbc$alkphos, main = "Serum Alk. Phos. vs. Serum Albumin",
+     xlab = "Serum Albumin", ylab = "Serum Alkaline Phosphatase",
+     pch = 19)
> plot(pbc$albumin, pbc$alkphos, main = "Serum Alk. Phos. vs. Serum Albumin",
+     xlab = label(pbc$albumin), ylab = label(pbc$alkphos), pch = 19)
```



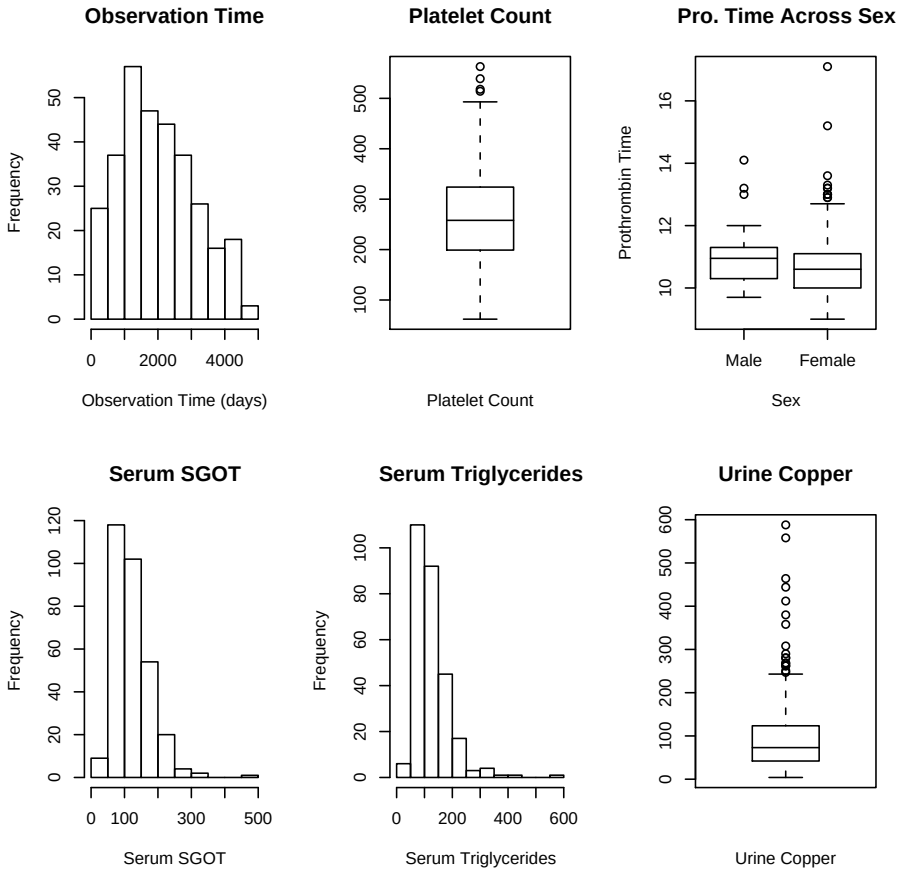

```
> plot(lowbw$age ~ lowbw$bwt, type = "n", main = "Mother's Age vs. Birth Weight",
+     xlab = "Mother's Weight", ylab = "Mother's Age")
> points(lowbw$bwt[lowbw$smoke == "No" & !is.na(lowbw$smoke)],
+     lowbw$age[lowbw$smoke == "No" & !is.na(lowbw$smoke)], pch = 19)
> points(lowbw$bwt[lowbw$smoke == "Yes" & !is.na(lowbw$smoke)],
+     lowbw$age[lowbw$smoke == "Yes" & !is.na(lowbw$smoke)], pch = 19,
+     col = "gray")
> abline(lm(age ~ bwt, data = lowbw[lowbw$smoke == "No" & !is.na(lowbw$smoke),
+     ]), col = "black")
> abline(lm(age ~ bwt, data = lowbw[lowbw$smoke == "Yes" & !is.na(lowbw$smoke),
+     ]), col = "gray")
```



NOTES: (1) `type = "n"` in the `plot` command will produce the axes, the axes labels, and the main title for the plot, but will not plot any of the points; (2) the `is.na` function indicates which elements are missing (i.e. `NA`), so `!is.na(lowbw$smoke)` will return only the non-missing values of `lowbw$smoke`; (3) the `abline` function adds a straight line to a plot

10.4 Multiple plots per page

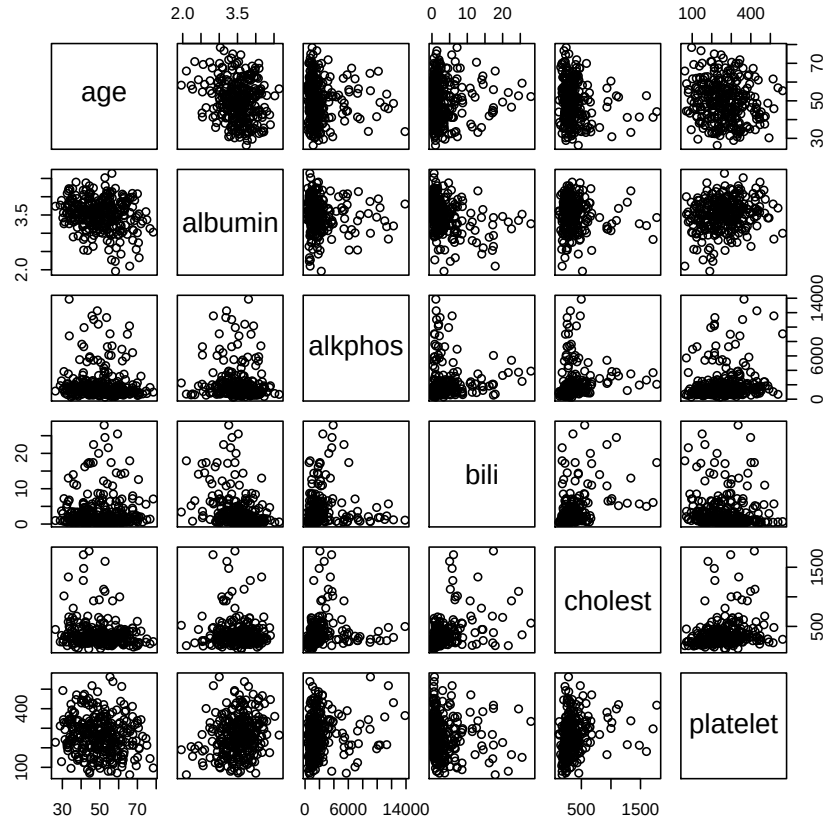
```
> par(mfrow = c(2, 3))
> hist(pbc$obstime, main = "Observation Time", xlab = "Observation Time (days)")
> boxplot(pbc$platelet, main = "Platelet Count", xlab = "Platelet Count")
> boxplot(pbc$protime ~ pbc$sex, main = "Pro. Time Across Sex",
+     xlab = "Sex", ylab = "Prothrombin Time")
> hist(pbc$sgot, main = "Serum SGOT", xlab = "Serum SGOT")
> hist(pbc$trig, main = "Serum Triglycerides", xlab = "Serum Triglycerides")
> boxplot(pbc$urinecu, main = "Urine Copper", xlab = "Urine Copper")
> par(mfrow = c(1, 1))
```



NOTES: (1) the `par` function is used to set graphical parameters; (2) the `mfrow` argument allows you to change the number of plots per page by specifying the number of rows, and the number of columns, respectively (by default, there is one plot per page, which is equivalent to one row and one column)

10.5 Pairs plots

```
> library(Hmisc)
> pairs(pbc[Cs(age, albumin, alkphos, bili, cholest, platelet)])
```



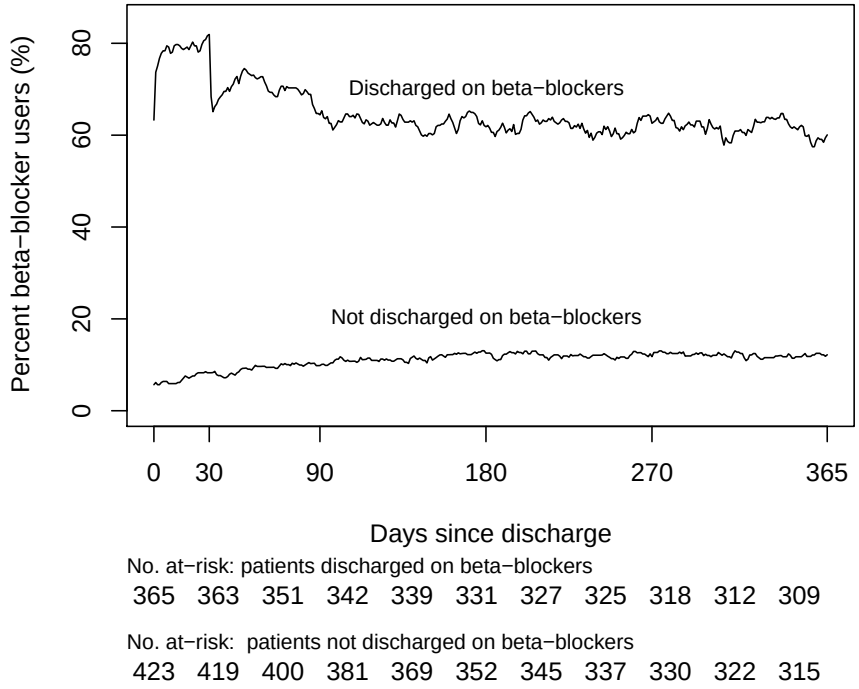
10.6 Graphs with text

`rates.dat` contains a dataset for a study of beta-blocker adherence post-AMI.

```
> rates <- read.table("rates.dat", header = T)
> par(oma = c(5, 0, 2, 0))
> plot(rates$day, rates$rate1, type = "l", ylim = c(0, 85), axes = F,
+     xlab = "Days since discharge", ylab = "Percent beta-blocker users (%)")
> axis(1, at = c(0, 30, 90, 180, 270, 365))
> axis(2, at = c(0, 20, 40, 60, 80))
> axis(1, at = rates$day, labels = rates$atrisk1, tick = F, line = 4,
+     cex = 0.8)
> axis(1, at = rates$day, labels = rates$atrisk0, tick = F, line = 6.5,
+     cex = 0.8)
> lines(rates$day, rates$rate0, type = "l")
> box()
```

```
> mtext("No. at-risk: patients discharged on beta-blockers", side = 1,
+     line = 4, adj = 0, cex = 0.8)
> mtext("No. at-risk:  patients not discharged on beta-blockers",
+     side = 1, line = 6.5, adj = 0, cex = 0.8)
> mtext("Figure 1.  Outpatient adherence to beta-blocker therapy post-AMI",
+     side = 3, cex = 1.2, line = 1)
> text(180, 70, "Discharged on beta-blockers", cex = 0.8)
> text(180, 20, "Not discharged on beta-blockers", cex = 0.8)
```

Figure 1. Outpatient adherence to beta−blocker therapy post−AM



NOTES: (1) the `oma` argument of the `par` function allows you to change the size of the outer margins of the plot given in lines of text (the order is bottom, left, top, right); (2) the `axis` function adds and axis to a plot, allowing the spefication of the side, position, label, and other options (corresponds to setting `axes = F` in the `plot` command); (3) the `lines` function adds a line to a plot; (4) the `box` function draws a box around a plot; (5) the `mtext` function writes text into the margins of a plot; and (6) the `text` function writes text inside a plot

10.7 Different page layouts

Unfortunately, the LaTex interface I used to create this pdf file had problems placing the following plot in this pdf file, but we can still run the code in R and view the resulting plot.

```
albhist<-hist(pbc$albumin, plot=FALSE)
obtimehist<-hist(pbc$obstime, plot=FALSE)
def.par <- par(no.readonly = TRUE) # save default, for resetting...
layout(matrix(c(2,0,1,3),2,2,byrow=TRUE), widths=c(3,1),
   heights=c(1,3), respect=TRUE)
plot(pbc$albumin, pbc$obstime, xlab="Serum Albumin",
   ylab="Observation Time", main="Observation Time by Serum Albumin")
barplot(albhist$counts, main="Serum Albumin",
   space=0)
barplot(obtimehist$counts, horiz=TRUE, main="Observation Time",
   space=0)
par(def.par)#- reset to default
```

10.8 Graphical Data Summary

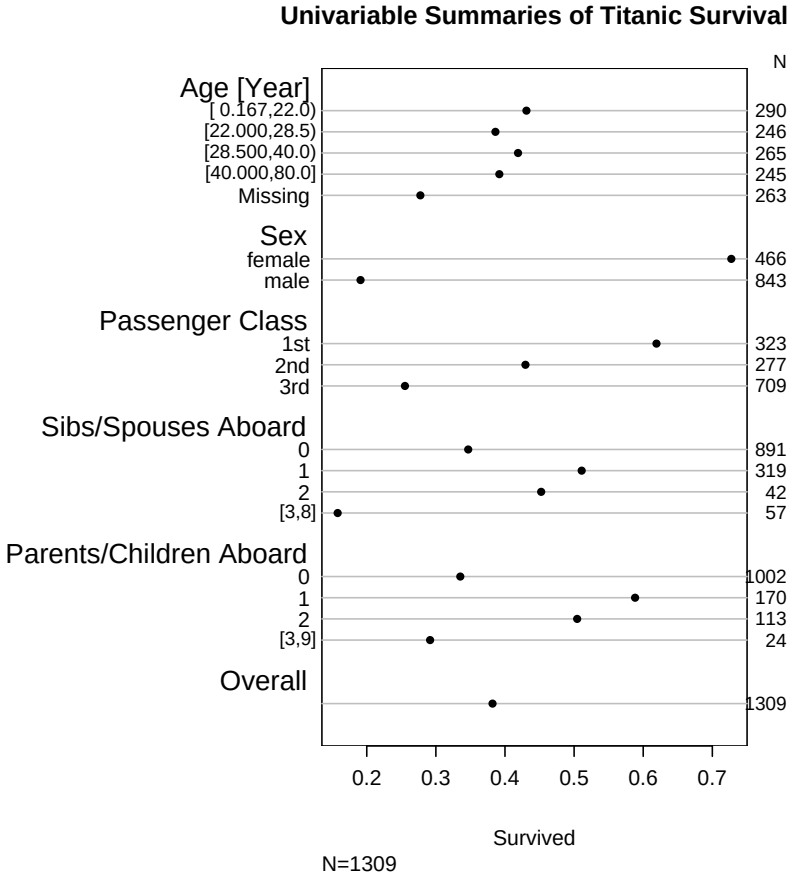
The `titanic3` data set contains information on N = 1309 passengers from the *Titanic*. Chapter 12 in Frank Harrell's *Regression Modelling Strategies* develops a binary logistic regression model to describe the patterns of survival in these passengers, based on passenger age, sex, ticket class, and the number of family members accompanying each passenger.[8]

```
> library(Hmisc)
> library(Design)
> getHdata(titanic3)
> x <- titanic3[Cs(pclass, survived, age, sex, sibsp, parch)]
> x <- upData(x, labels = c(sex = "Sex", pclass = "Passenger Class",
+     sibsp = "Sibs/Spouses Aboard", parch = "Parents/Children Aboard"))

Input object size:         54396 bytes;         6 variables
New object size:          54604 bytes;         6 variables

> dd <- datadist(x)
> options(datadist = "dd")
> titanic.summ <- summary(survived ~ age + sex + pclass + cut2(sibsp,
+     0:3) + cut2(parch, 0:3), data = x)

> latex(titanic.summ, file = "")


> par(cex = 0.85)
> plot(titanic.summ, main = "Univariable Summaries of Titanic Survival")
```

[8] *Regression Modelling Strategies*, Harrell

Table 3: Survived N=1309

		N	survived
Age	Year		
[0.167,22.0)		290	0.43
[22.000,28.5)		246	0.39
[28.500,40.0)		265	0.42
[40.000,80.0]		245	0.39
Missing		263	0.28
Sex			
female		466	0.73
male		843	0.19
Passenger Class			
1st		323	0.62
2nd		277	0.43
3rd		709	0.26
Sibs/Spouses Aboard			
0		891	0.35
1		319	0.51
2		42	0.45
[3,8]		57	0.16
Parents/Children Aboard			
0		1002	0.34
1		170	0.59
2		113	0.50
[3,9]		24	0.29
Overall			
		1309	0.38



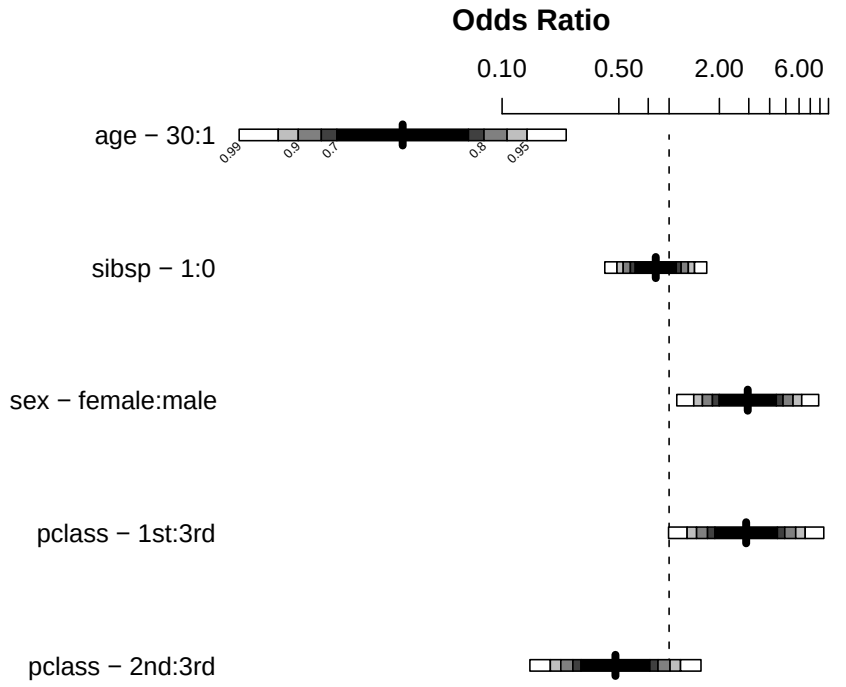
10.9 Summarizing/Describing the Fitted Model

It is always important for the analyst to present and interpret a fitted model, once the proper variables have been modelled and all assumptions have been met. The coefficients in the model may be interpreted by computing, for each variable, the change in log odds for a sensible change in the variable value (e.g. interquartile range).[9]

```
> library(Hmisc)
> library(Design)
> dd <- datadist(x)
> options(datadist = "dd")
> titanic.model <- lrm(survived ~ (sex + pclass + rcs(age, 5))^2 +
+     rcs(age, 5) * sibsp, data = x)
```

[9] *Regression Modelling Strategies*, Harrell

```
> titanic.model.summ <- summary(titanic.model, age = c(1, 30),
+     sibsp = 0:1)

> plot(titanic.model.summ, log = T)
```



11 Writing Your Own Functions

- Perhaps one of the best features of R is its capability of writing your own functions
- Writing your own functions will become very useful when you find yourself executing the same set of commands (e.g. finding the mean and standard deviation, or plotting the same general plot) repeated times
- ***Some Examples:***

1. A function that prints out the mean and standard deviation of a set of numbers:[10]

```
> mean.and.sd <- function(x) {
+     av <- mean(x)
+     stdev <- sd(x)
+     c(mean = av, SD = stdev)
+ }
> mean.and.sd(1:10)
```

```
    mean       SD
5.500000 3.027650
```

```
> mean.and.sd(lowbw$age)
```

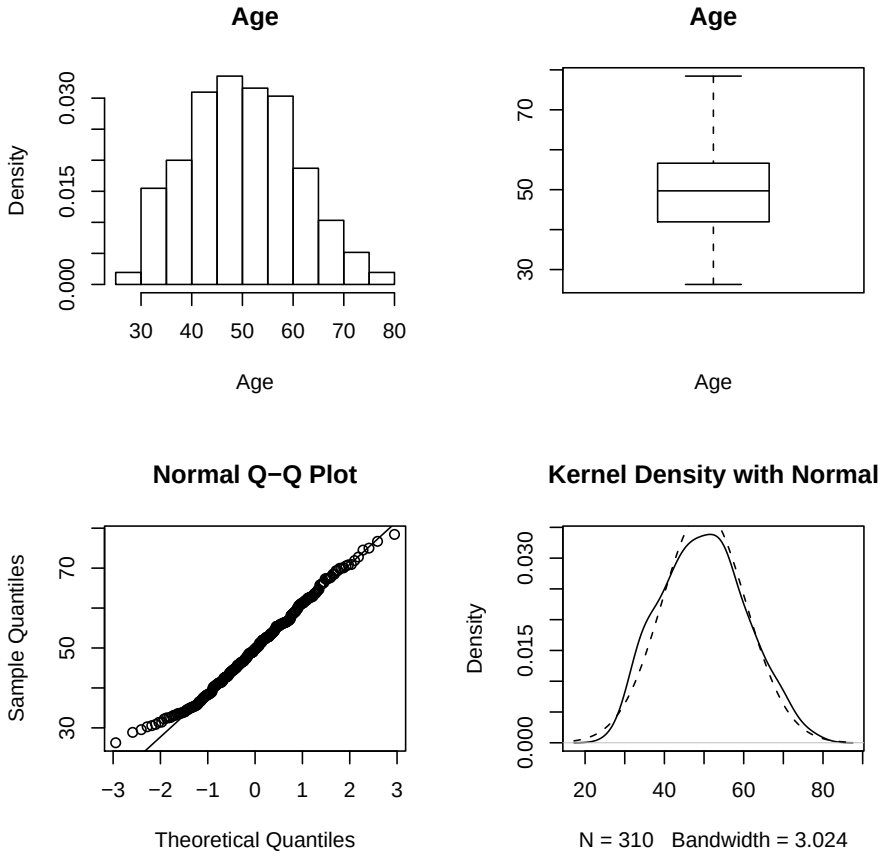
```
     mean        SD
23.238095  5.298678
```

2. A functions that generates four plots for a continuous variable: (1) a histogram, (2) a boxplot, (3) a normal Q-Q plot, and (4) a Kernel Density plot:

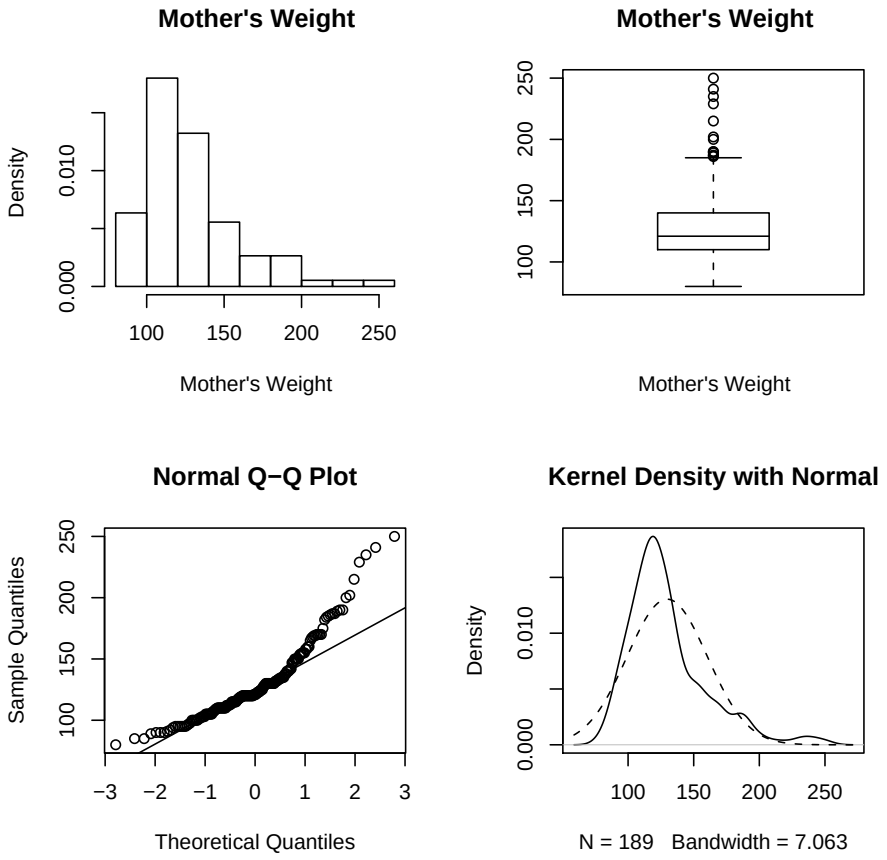
```
> dist.shape <- function(xdata, varname) {
+     par(mfrow = c(2, 2))
+     hist(xdata, freq = F, main = varname, xlab = varname)
+     boxplot(xdata, main = varname, xlab = varname)
+     qqnorm(xdata)
+     qqline(xdata)
+     dx <- density(xdata)
+     xbar <- mean(xdata)
+     stdev <- sd(xdata)
+     rangex <- range(dx$x)
+     xx <- seq(rangex[1], rangex[2], length = length(dx$y))
+     plot(dx, main = "Kernel Density with Normal")
+     lines(xx, dnorm(xx, xbar, stdev), lty = 2)
+ }
```

```
> dist.shape(pbc$age, "Age")
```

[10] *Data Analysis and Graphics Using R*, Maindonald and Braun




```
> dist.shape(lowbw$lwt, "Mother's Weight")
```



3. A function that finds high and low outliers (greater than +/-4 S.D. of the mean) of one or more continuous variables (assumes first column of data frame represents an ID):

```
> out <- function(dataframe, colnames, idname) {
+     data <- dataframe[, colnames]
+     I <- dim(data)[2]
+     for (i in 1:I) {
+         colm <- data[!is.na(data[[i]]), i]
+         colname <- colnames[[i]]
+         con <- 4
+         bottom <- mean(colm) - con * sd(colm)
+         top <- mean(colm) + con * sd(colm)
+         nhigh <- length(colm[colm > top])
+         nlow <- length(colm[colm < bottom])
+         if (nlow > 0) {
+             lowouts <- dataframe[colm < bottom & !is.na(data[[i]]),
```

```
+                 c(idname, colname)]
+             cat("LOW OUTLIERS", "\t", "( <", bottom, ")", "\n",
+                 "ID", "\t", colname, "\n", file = "outliers.txt",
+                 append = T)
+             J <- dim(lowouts)[1]
+             K <- dim(lowouts)[2]
+             for (j in 1:J) {
+                 for (k in 1:K) {
+                   cat(paste(lowouts[j, k]), "\t", file = "outliers.txt",
+                     append = T)
+                   if (k == K)
+                     cat("\n", file = "outliers.txt", append = T)
+                 }
+             }
+             cat("-------------------------------------", "\n",
+                 "\n", file = "outliers.txt", append = T)
+         }
+         if (nhigh > 0) {
+             highouts <- dataframe[colm > top & !is.na(data[[i]]),
+                 c(idname, colname)]
+             cat("HIGH OUTLIERS", "\t", "( >", top, ")", "\n",
+                 "ID", "\t", colname, "\n", file = "outliers.txt",
+                 append = T)
+             J <- dim(highouts)[1]
+             K <- dim(highouts)[2]
+             for (j in 1:J) {
+                 for (k in 1:K) {
+                   cat(paste(highouts[j, k]), "\t", file = "outliers.txt",
+                     append = T)
+                   if (k == K)
+                     cat("\n", file = "outliers.txt", append = T)
+                 }
+             }
+             cat("-------------------------------------", "\n",
+                 "\n", file = "outliers.txt", append = T)
+         }
+     }
+ }
> out(lowbw, c("age", "lwt", "bwt"), "id")
```

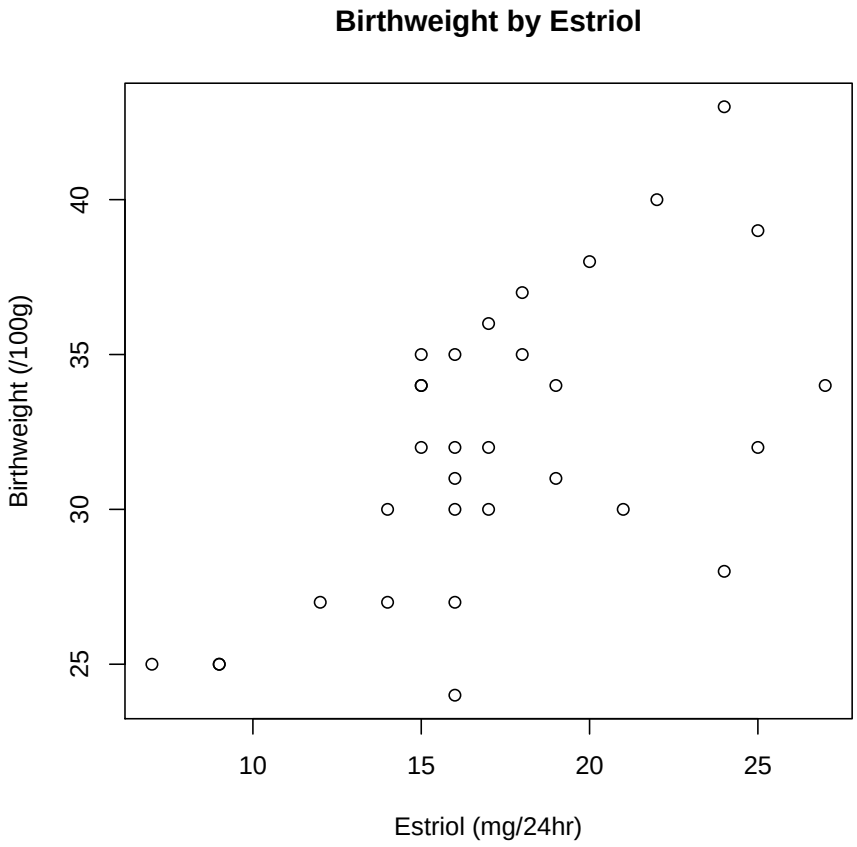
12 Statistics with R

12.1 Correlation

Example: **Estriol and Birthweight**

Consider a study investigating birthweight (/100 g) and estriol levels (mg/24hr) in pregnant women (estriol.dta)

```
> library(foreign)
> est <- read.dta("estriol.dta")
> names(est)

[1] "estriol" "birthwt"

> dim(est)

[1] 31  2

> plot(est$estriol, est$birthwt, main = "Birthweight by Estriol",
+     xlab = "Estriol (mg/24hr)", ylab = "Birthweight (/100g)")
```



12.1.1 Pearson Correlation & Testing for Association

```
> cor(est)

          estriol   birthwt
estriol 1.0000000 0.6097313
birthwt 0.6097313 1.0000000

> cor.test(est$estriol, est$birthwt)

        Pearson's product-moment correlation

data:  est$estriol and est$birthwt
t = 4.1427, df = 29, p-value = 0.0002712
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.3257757 0.7927878
sample estimates:
      cor
0.6097313
```

12.1.2 Spearman Rank Correlation

```
> library(Hmisc)
> rcorr(est$estriol, est$birthwt, type = "spearman")

     x    y
x 1.00 0.56
y 0.56 1.00

n= 31


P
  x     y
x       0.001
y 0.001

> rcorr(est$estriol, est$birthwt, type = "pearson")

     x    y
x 1.00 0.61
y 0.61 1.00

n= 31


P
```

```
  x     y
x       3e-04
y 3e-04
```

12.2 Simple Linear Regression

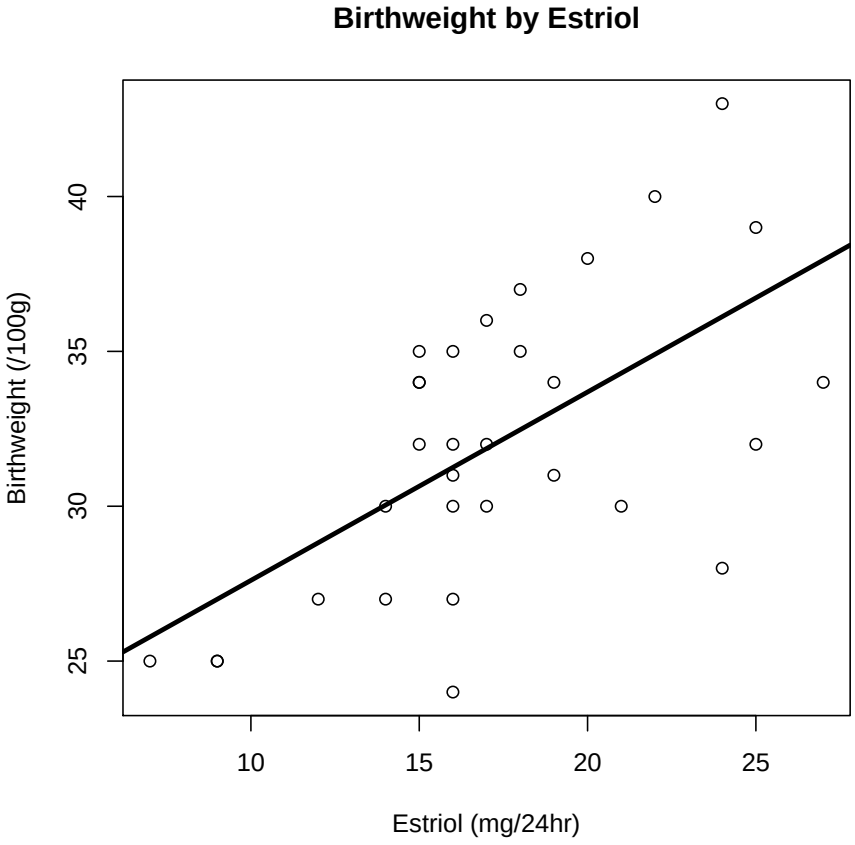
Example: **Estriol and Birthweight**

Consider regressing birthweight on estriol.

```
> est.m <- lm(birthwt ~ estriol, data = est)
> summary(est.m)

Call:
lm(formula = birthwt ~ estriol, data = est)

Residuals:
     Min       1Q   Median       3Q      Max
-8.12000 -2.03810 -0.03810  3.35371  6.88000

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  21.5234     2.6204   8.214 4.68e-09 ***
estriol       0.6082     0.1468   4.143 0.000271 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.821 on 29 degrees of freedom
Multiple R-Squared: 0.3718,        Adjusted R-squared: 0.3501
F-statistic: 17.16 on 1 and 29 DF,  p-value: 0.0002712
```

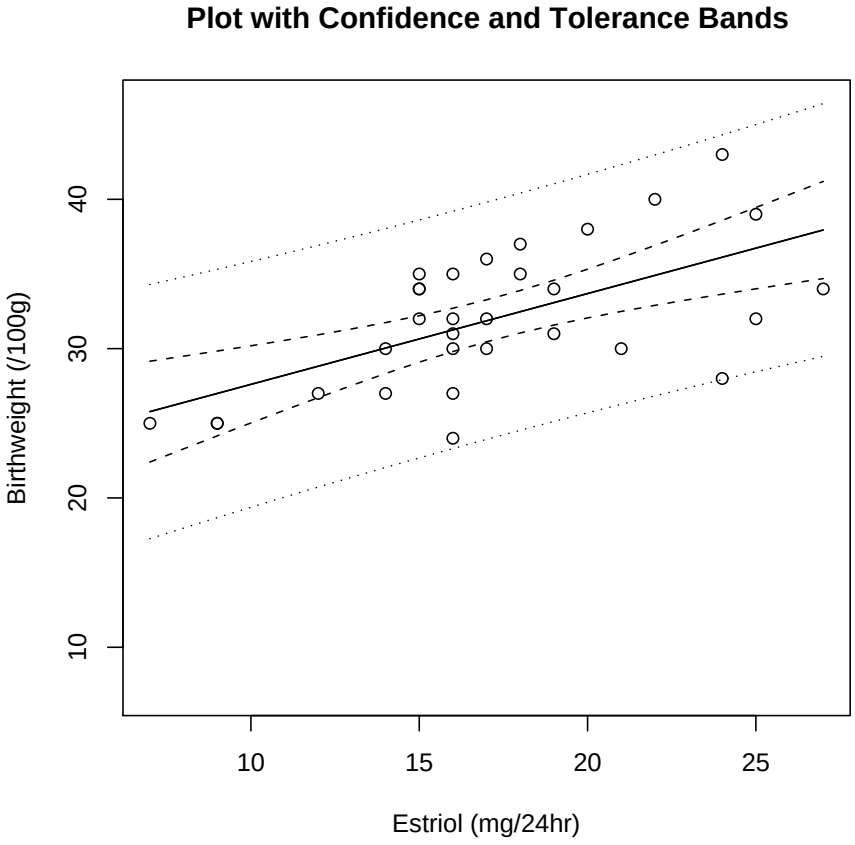
12.2.1 Interpretation of Output

```
> plot(est$estriol, est$birthwt, main = "Birthweight by Estriol",
+     xlab = "Estriol (mg/24hr)", ylab = "Birthweight (/100g)")
> abline(est.m, lwd = 3)
```



12.2.2 Plotting 95% Confidence Intervals & Prediction Intervals

```
> pred.frame <- data.frame(estriol = 7:27)
> pp <- predict(est.m, int = "p", newdata = pred.frame)
> pc <- predict(est.m, int = "c", newdata = pred.frame)
> pred.estriol <- pred.frame$estriol

> plot(est$estriol, est$birthwt, ylim = range(est$estriol, pp,
+     na.rm = T), main = "Plot with Confidence and Tolerance Bands",
+     xlab = "Estriol (mg/24hr)", ylab = "Birthweight (/100g)")
> matlines(pred.estriol, pc, lty = c(1, 2, 2), col = "black")
> matlines(pred.estriol, pp, lty = c(1, 3, 3), col = "black")
```



12.2.3 Linear Regression for Categorical Variables

Example: **Low Birthweight Study**

We would like to determine whether there is a difference in mean birthweight for those with no previous pre-term labor and those with 1+ pre-term labors.

```
> lowbw$ptl.cat <- ifelse(lowbw$ptl > 0, 1, 0)
> names(lowbw)

 [1] "id"      "low"     "age"     "lwt"     "race"    "smoke"   "ptl"
 [8] "ht"      "ui"      "ftv"     "bwt"     "ptl.cat"

> table(lowbw$ptl)

  0   1   2   3
159  24   5   1
```

```
> table(lowbw$ptl.cat)

  0   1
159  30

> lowbw.m <- lm(bwt ~ ptl.cat, data = lowbw)
> summary(lowbw.m)

Call:
lm(formula = bwt ~ ptl.cat, data = lowbw)

Residuals:
Birthweight [grams]
      Min        1Q    Median        3Q       Max
-1992.572  -495.400    -8.572   586.428  1976.428

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  3013.57      56.57  53.269  < 2e-16 ***
ptl.cat      -434.17     142.00  -3.058  0.00256 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 713.4 on 187 degrees of freedom
Multiple R-Squared: 0.04761,        Adjusted R-squared: 0.04252
F-statistic: 9.349 on 1 and 187 DF,  p-value: 0.002558

> t.test(bwt ~ ptl.cat, data = lowbw, var.equal = T, alternative = "two.sided")

        Two Sample t-test

data:  bwt by ptl.cat
t = 3.0576, df = 187, p-value = 0.002558
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 154.0518 714.2929
sample estimates:
mean in group 0 mean in group 1
       3013.572        2579.400

> t.test(bwt ~ ptl.cat, data = lowbw, var.equal = T, alternative = "less")

        Two Sample t-test

data:  bwt by ptl.cat
t = 3.0576, df = 187, p-value = 0.9987
alternative hypothesis: true difference in means is less than 0
```

```
95 percent confidence interval:
     -Inf 668.8983
sample estimates:
mean in group 0 mean in group 1
       3013.572        2579.400
```

```
> t.test(bwt ~ ptl.cat, data = lowbw, var.equal = T, alternative = "greater")

        Two Sample t-test

data:  bwt by ptl.cat
t = 3.0576, df = 187, p-value = 0.001279
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 199.4463      Inf
sample estimates:
mean in group 0 mean in group 1
       3013.572        2579.400
```

12.3 Simple Logistic Regression

Example: **Estriol and Birthweight**

Suppose we wish to relate estriol level to the probability of having a low birthweight infant, where a low birthweight indicator in which birthweight <3000 grams indicates low birthweight.

```
> est$birthwt.ind <- ifelse(est$birthwt < 30, 1, 0)
> table(est$birthwt.ind)

 0  1
23  8
```

```
> library(Hmisc)
> library(Design)
> dd <- datadist(est)
> options(datadist = "dd")
> est.birthwt.m <- lrm(birthwt.ind ~ estriol, data = est)
> summary(est.birthwt.m)

             Effects              Response : birthwt.ind

 Factor      Low High Diff. Effect S.E. Lower 0.95 Upper 0.95
 estriol     15  19.5 4.5   -1.56  0.7  -2.93      -0.20
  Odds Ratio 15  19.5 4.5    0.21   NA   0.05       0.82
```

```
> anova(est.birthwt.m)
```

```
               Wald Statistics          Response: birthwt.ind

 Factor     Chi-Square d.f. P
 estriol    5.03       1    0.0249
 TOTAL      5.03       1    0.0249
```

12.4 Simple Proportional Hazards Regression

Example: **Primary Biliary Cirrhosis (PBC) Trial**

We would like to investigate the role of d-penicillamine (DPCA) for treating PBC on the patients' survival.

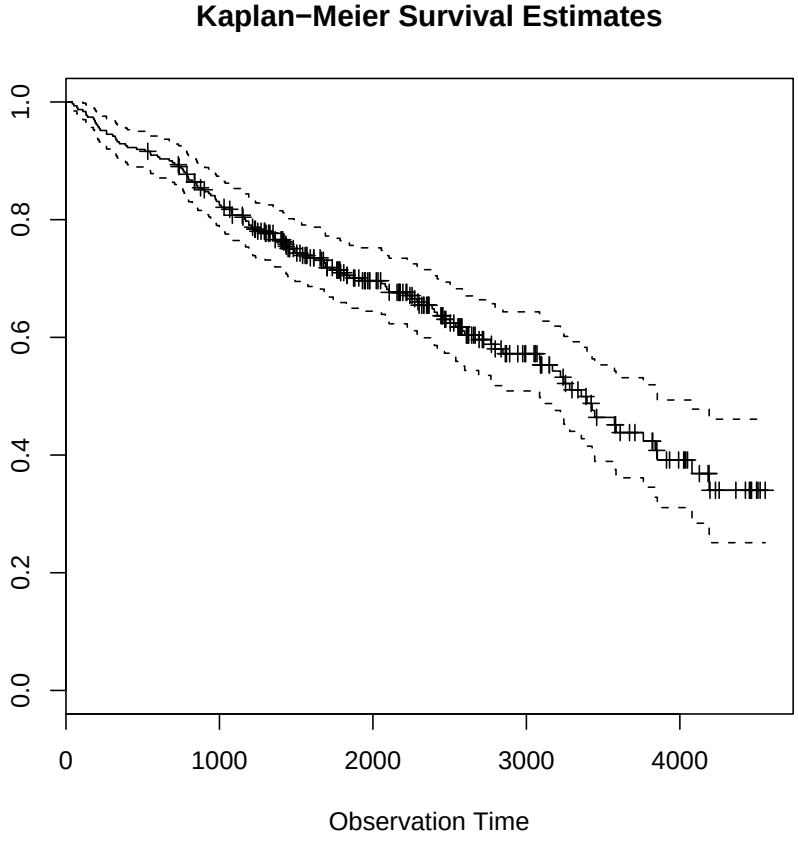
12.4.1 Kaplan-Meier Estimates

First lets just look at patient survival, regardless of treatment.

```
> library(survival)

> pbc.surv <- survfit(Surv(obstime, status == "Died"), data = pbc)
> pbc.surv

Call: survfit(formula = Surv(obstime, status == "Died"), data = pbc)

       n    events     rmean se(rmean)    median   0.95LCL   0.95UCL
     310       125      2971       102      3358      3086      3853

> summary(pbc.surv)

Call: survfit(formula = Surv(obstime, status == "Died"), data = pbc)

 time n.risk n.event survival std.err lower 95% CI upper 95% CI
   41    310       1    0.997 0.00322        0.990        1.000
   51    309       1    0.994 0.00455        0.985        1.000
   71    308       1    0.990 0.00556        0.979        1.000
   77    307       1    0.987 0.00641        0.975        1.000
  110    306       1    0.984 0.00715        0.970        0.998
  130    305       1    0.981 0.00782        0.965        0.996
  131    304       1    0.977 0.00844        0.961        0.994
  140    303       1    0.974 0.00901        0.957        0.992
  179    302       1    0.971 0.00954        0.952        0.990
  186    301       1    0.968 0.01004        0.948        0.988
.
.
.
 3395     43       1    0.488 0.04022        0.415        0.573
 3428     41       1    0.476 0.04096        0.402        0.563
 3445     40       1    0.464 0.04163        0.389        0.553
```

```
 3574     37       1    0.451 0.04235        0.376        0.543
 3584     34       1    0.438 0.04314        0.361        0.531
 3762     30       1    0.424 0.04410        0.345        0.519
 3839     27       1    0.408 0.04517        0.328        0.507
 3853     25       1    0.392 0.04622        0.311        0.494
 4079     17       1    0.369 0.04891        0.284        0.478
 4191     13       1    0.340 0.05272        0.251        0.461
```

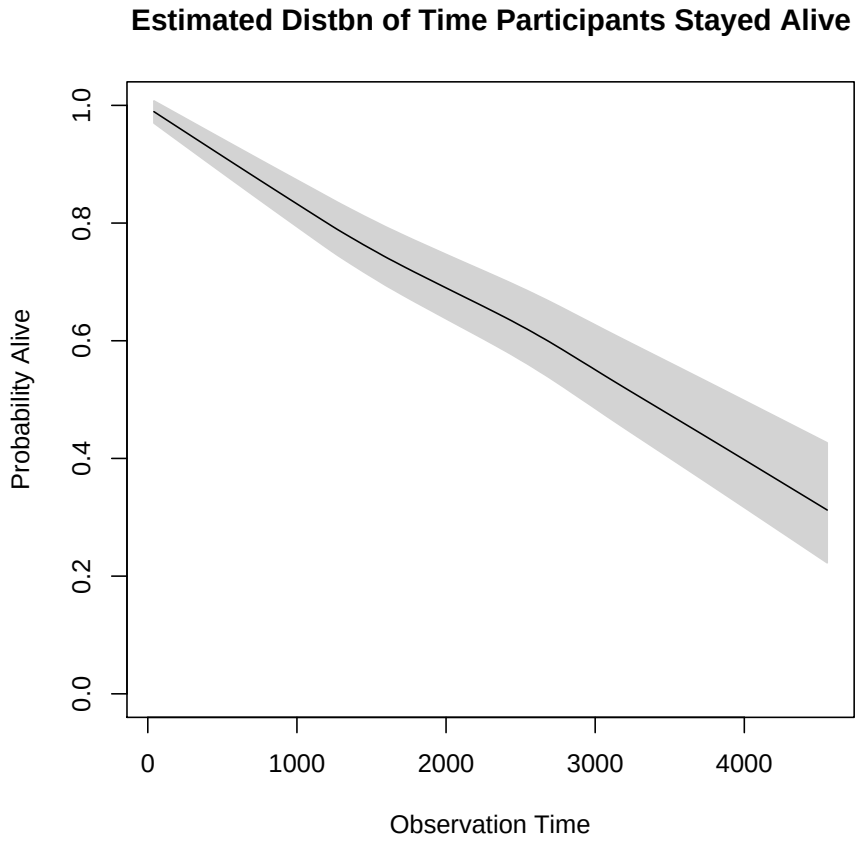
```
> plot(pbc.surv, main = "Kaplan-Meier Survival Estimates", xlab = "Observation Time")
```



The following plot shows the estimated (smoothed) distribution of time patients stayed alive. The shaded area represents the 95% confidence interval. A similar plot was shown in an article published in the Journal of the American Medical Association (*JAMA, June 20, 2001 – Vol 285, No. 23*).

```
> xx <- c(pbc.surv$time, rev(pbc.surv$time))
> yy <- c(lowess(pbc.surv$time, pbc.surv$upper)$y, rev(lowess(pbc.surv$time,
+     pbc.surv$lower)$y))
```

```
> plot(pbc.surv$time, pbc.surv$surv, type = "n", ylim = c(0, 1),
+     xlab = "Observation Time", ylab = "Probability Alive", main = "Estimated Distbn of Tim
> polygon(xx, yy, col = "lightgray", border = "lightgray")
> lines(lowess(pbc.surv$time, pbc.surv$surv))
```

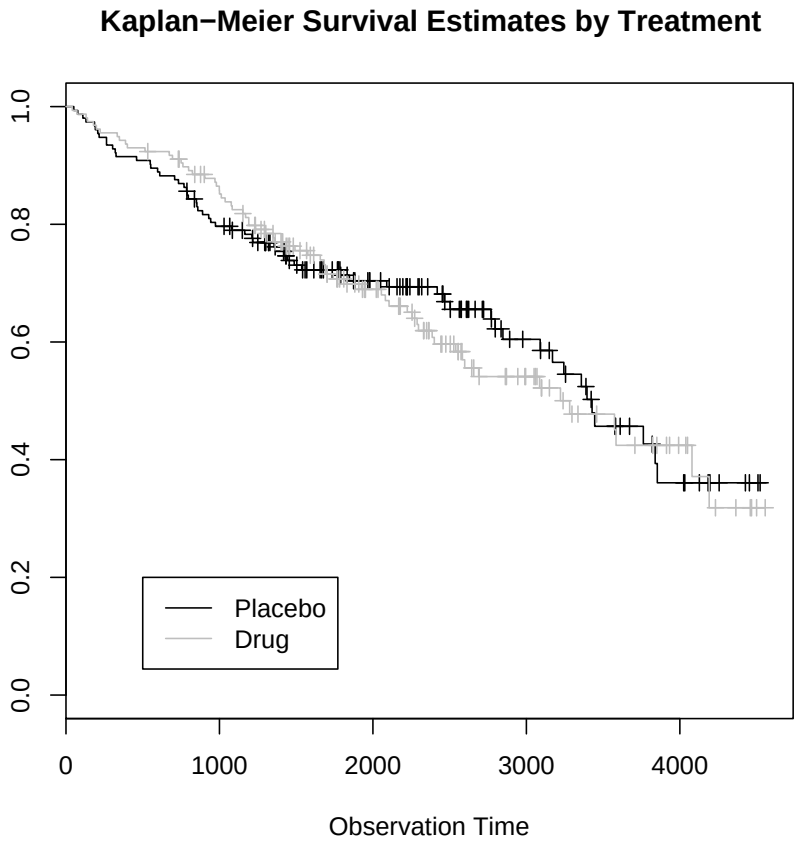


Now let's look at the role of treatment on patient survival.

```
> pbc.surv.tx <- survfit(Surv(obstime, status == "Died") ~ tx,
+     data = pbc)
> pbc.surv.tx

Call: survfit(formula = Surv(obstime, status == "Died") ~ tx, data = pbc)

             n events rmean se(rmean) median 0.95LCL 0.95UCL
tx=Placebo 153     60  2987       144   3428    3090     Inf
tx=Drug    157     65  2947       142   3282    2583     Inf

> summary(pbc.surv.tx)
```

```
Call: survfit(formula = Surv(obstime, status == "Died") ~ tx, data = pbc)

               tx=Placebo
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
   51    153       1    0.993 0.00651        0.981        1.000
   77    152       1    0.987 0.00918        0.969        1.000
  110    151       1    0.980 0.01121        0.959        1.000
  130    150       1    0.974 0.01290        0.949        0.999
  186    149       1    0.967 0.01437        0.940        0.996
.
.
.
 3428     22       1    0.480 0.05977        0.376        0.612
 3445     21       1    0.457 0.06113        0.351        0.594
 3762     15       1    0.426 0.06419        0.317        0.573
 3839     13       1    0.393 0.06711        0.282        0.550
 3853     12       1    0.361 0.06907        0.248        0.525

               tx=Drug
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
   41    157       1    0.994 0.00635        0.981        1.000
   71    156       1    0.987 0.00895        0.970        1.000
  131    155       1    0.981 0.01093        0.960        1.000
  140    154       1    0.975 0.01258        0.950        0.999
  179    153       1    0.968 0.01401        0.941        0.996
.
.
.
 3282     22       1    0.477 0.05492        0.381        0.598
 3574     18       1    0.451 0.05792        0.351        0.580
 3584     17       1    0.424 0.06028        0.321        0.561
 4079      8       1    0.371 0.07242        0.253        0.544
 4191      7       1    0.318 0.07915        0.195        0.518
```

```
> plot(pbc.surv.tx, col = c("black", "gray"), main = "Kaplan-Meier Survival Estimates by Tre
+     xlab = "Observation Time")
> legend(500, 0.2, legend = c("Placebo", "Drug"), col = c("black",
+     "gray"), lty = c(1, 1))
```

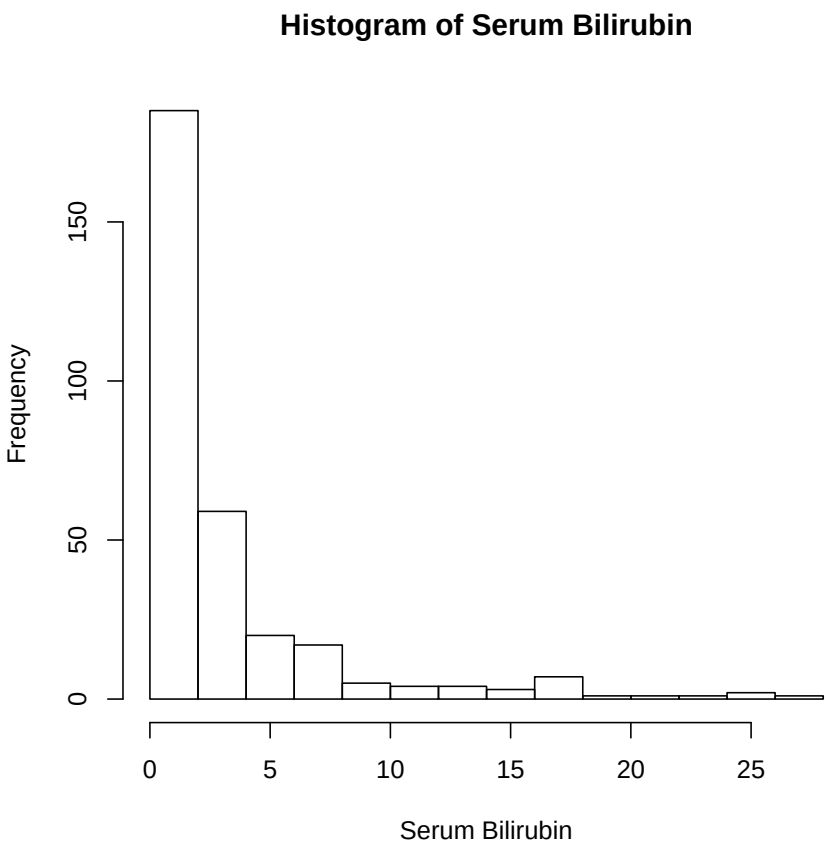


12.4.2 Log-Rank Test

```
> library(survival)
> survdiff(Surv(pbc$obstime, pbc$status == "Died") ~ pbc$tx)

Call:
survdiff(formula = Surv(pbc$obstime, pbc$status == "Died") ~
    pbc$tx)

                N Observed Expected (O-E)^2/E (O-E)^2/V
pbc$tx=Placebo 153       60     61.8    0.0507     0.100
pbc$tx=Drug    157       65     63.2    0.0495     0.100

 Chisq= 0.1  on 1 degrees of freedom, p= 0.751
```

12.4.3 Cox Proportional Hazards Regression

```
> library(survival)
> pbc.coxph.tx <- coxph(Surv(obstime, status == "Died") ~ tx, data = pbc)
> summary(pbc.coxph.tx)

Call:
coxph(formula = Surv(obstime, status == "Died") ~ tx, data = pbc)

  n= 310

         coef exp(coef) se(coef)     z    p
txDrug 0.0568      1.06    0.179 0.317 0.75

       exp(coef) exp(-coef) lower .95 upper .95
txDrug      1.06      0.945     0.745      1.50

Rsquare= 0   (max possible= 0.984 )
Likelihood ratio test= 0.1  on 1 df,   p=0.751
Wald test            = 0.1  on 1 df,   p=0.751
Score (logrank) test = 0.1  on 1 df,   p=0.751

> hist(pbc$bili, main = "Histogram of Serum Bilirubin", xlab = "Serum Bilirubin")
```




```
> pbc.coxph.logbili <- coxph(Surv(obstime, status == "Died") ~
+     log(bili), data = pbc)
> summary(pbc.coxph.logbili)

Call:
coxph(formula = Surv(obstime, status == "Died") ~ log(bili),
    data = pbc)

  n= 310

          coef exp(coef) se(coef)    z p
log(bili) 1.08      2.96   0.0934 11.6 0

          exp(coef) exp(-coef) lower .95 upper .95
log(bili)      2.96      0.338      2.46      3.55

Rsquare= 0.348   (max possible= 0.984 )
```

```
Likelihood ratio test= 133  on 1 df,   p=0
Wald test            = 135  on 1 df,   p=0
Score (logrank) test = 159  on 1 df,   p=0
```