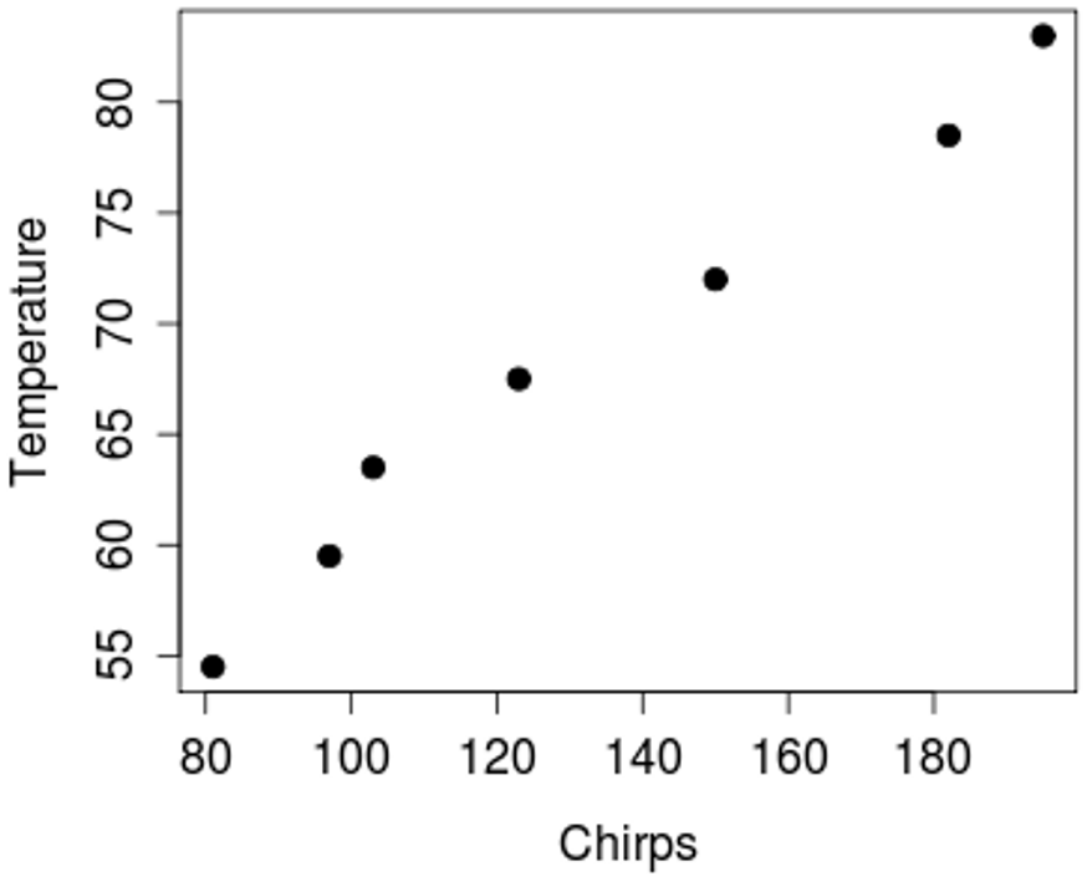
Lecture 2: Introduction to Linear Regression

Ailin Zhang

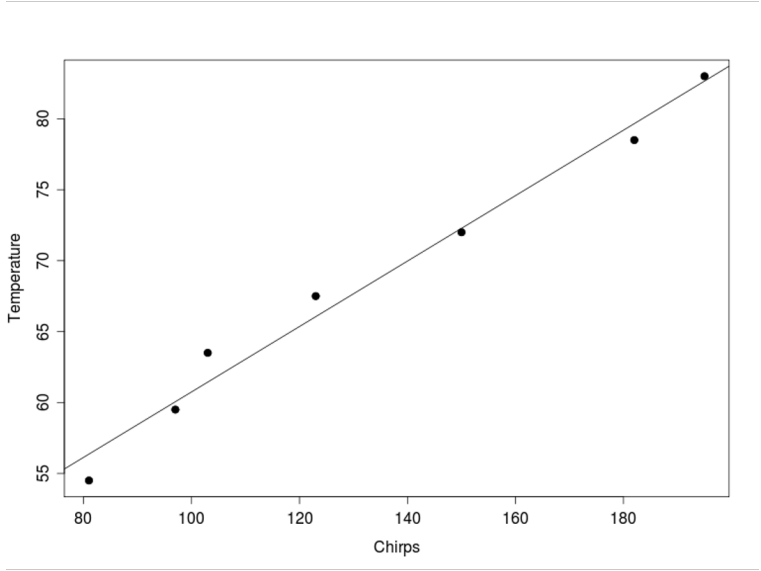
2023-05-10

Can you estimate the temperature by listening to cricket's chirp?



Regression Line

Goal: Find a straight line that best fits the data in the scatterplot



Data: values $(x_1, y_1), \ldots, (x_n, y_n)$ of (X, Y) observed on each of n units or cases. The estimated regression line is:

$$\hat{y} = \underbrace{b_1}_{\text{slope}} x + \underbrace{b_0}_{\text{intercept}}$$

where x is the explanatory variable, and $\hat{y}$ is the predicted response variable.

Interpretation of slope and intercept

- Slope: increase in predicted y for every unit increase in x
- Intercept: predicted y value when x = 0

$\hat{Temp} = 37.68 + 0.23 Chirps$

Which is a correct interpretation?

1. The average temperature is $37.68°F$
2. For every extra 0.23 chirps per minute, the predicted temperate increases by $1°F$
3. Predicted temperature increases by 0.23 degrees for each extra chirp per minute
4. For every extra 0.23 chirps per minute, the predicted temperature increases by $37.68°F$

Simple Linar Regression

When there is only one predictor X. The model is called simple linear regression model.

$$y = \beta_0 + \beta_1 x + \epsilon$$

- β_0 and β_1 are unknown population parameters, therefore are estimated from the data.
- ϵ is a random variable to represent noise/ uncertainty.

We can also write the simple linear regression equation as:

$$E(Y|X = x) = \beta_0 + \beta_1 x$$

- $E(Y|X = x)$ is the expected value of Y for a given x value.

Assumptions about ϵ

$$\epsilon_i \sim N(0, \sigma^2)$$

1. The error ϵ is a random variable with mean of zero.
2. The variance of ϵ, denoted by σ^2, is the same for all values of the independent variable. (Homoscedasticity)
3. ϵ is independent.
4. The error ϵ is a normally distributed random variable.

Estimated Simple Linear Regression Equation

The estimated simple linear regression equation: $\hat{y} = b_0 + b_1 x$

Compare with regression equation: $E(Y|X = x) = \beta_0 + \beta_1 x$

b_0 and b_1 provide estimates of β_0 and β_1

	Population parameter	Sample statistic
Intercept	β_0	b_0
Slope	β_1	b_1
Equation	$E(Y\|X = x) = \beta_0 + \beta_1 x$	$\hat{y} = b_0 + b_1 x$

Linear Regression Models

$$Y = \beta_0 + \beta_1 X1 + \beta_2 X2 + \cdots + \beta_p X_p + \epsilon$$

More generally, linear regression can be categorized as a model is linear in its parameters $\beta_0, \beta_1, \ldots, \beta_p$. For example, the following are linear regression models:

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \epsilon$$

$$Y = \beta_0 + \beta_1 \log(X) + \epsilon$$

even though the relationship between Y and X is not linear.

- Linear in parameters, not linear in predictors
- less restrictive than you might think

Some Non-linear Models Can Be Turned Linear

EX 1:

$$Y = \frac{X}{\alpha X + \beta}$$

Take the reciprocal:

$$1/Y = \alpha + \beta(1/X)$$

Linear model:

$$Y' = \alpha + \beta X'$$

EX 2: Can the following model be turned into a linear model?

$$V = \alpha \times r_1^{\beta} \times h^{\beta_2}$$

Which of the Following Models are Linear?

(a) $Y = \beta_0 + \beta_1 X + \epsilon$

(b) $Y = \beta_0 \beta_1^X \epsilon$

(c) $Y = \beta_0 + \beta_1 e^X + \epsilon$

(d) $Y = \beta_0 + \beta_1 X^2 + \beta_2 \log(X) + \epsilon$

Answer: (c) and (d)

Which of the following models can be turned linear after transformation?

Multiple Linear Regression Model

	SLR		MLR				
	X	Y	X_1	X_2	$\dots$	X_p	Y
case 1:	x_1	y_1	x_{11}	x_{12}	$\dots$	x_{1p}	y_1
case 2:	x_2	y_2	x_{21}	x_{22}	$\dots$	x_{2p}	y_2
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
case n:	x_n	y_n	x_{n1}	x_{n2}	$\dots$	x_{np}	y_n

- In simple linear regression (SLR), we observe one predictor X.
- In multiple linear regression (MLR), we observe p predictors (explanatory variables, covariates)
- Each row is called a case, a record, or a data point
- y_i is the response (or dependent variable) of the ith case
- x_{ik} is the value of the explanatory variable X_k of the ith case

Multiple Linear Regression Models

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \epsilon_i$$

- $\epsilon_i \overset{\text{i.i.d.}}{\sim} N(0, \sigma^2)$
- Parameters included
  - β_0: intercept
  - β_k: regression coefficient (slope) for the kth explanatory variable
  - σ^2: variance of errors
- Observed (known): $y_i, x_{i1}, \ldots, x_{ip}$

  Unknown: $\beta_0, \beta_1, \ldots, \beta_p, \sigma^2, \epsilon_i$
- Random: ϵ_i, y_i

  Constants (not random): $\beta_k, \sigma^2, x_{ik}$

Multiple Linear Regression Models in Matrix Notation

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n1} & \dots & x_{np} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

Or

$$Y = X\beta + \epsilon$$

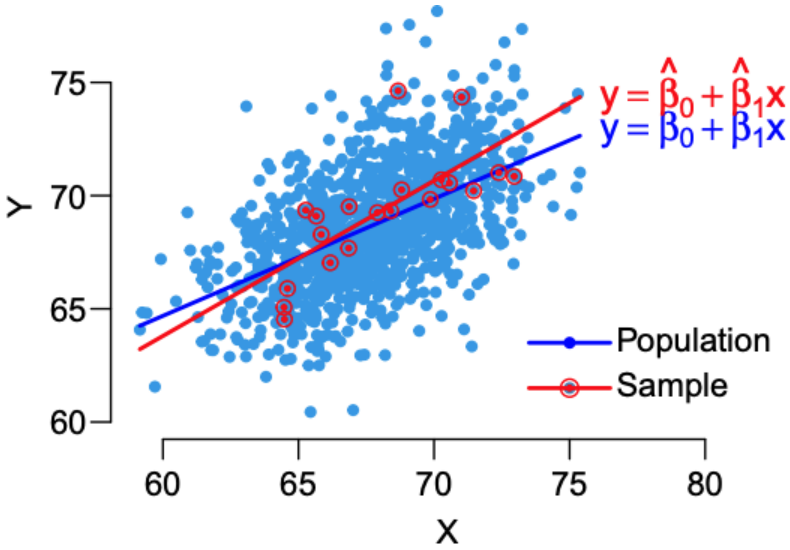
Estimated Multiple Linear Regression Equation

Hat notation is used:

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_p x_{ip}$$

- $\hat{y}_i$ is the fitted value
- y_i is the actual observed value

Parameters v.s. Estimates



$y = \beta_0 + \beta_1 x$	$y = \hat{\beta}_0 + \hat{\beta}_1 x$
Regression line of the population	Regression line of the sample
fixed	random, changes from sample to sample
unknown	can be calculated from sample
of interest	not of interest

Errors and Residuals

- Error (ϵ_i) can not be directly computed,
  $\epsilon_i = y_i - \beta_0 - \beta_1 x_{i1} - \cdots - \beta_p x_{ip}$
- The errors ϵ_i can be estimatedby residuals e_i

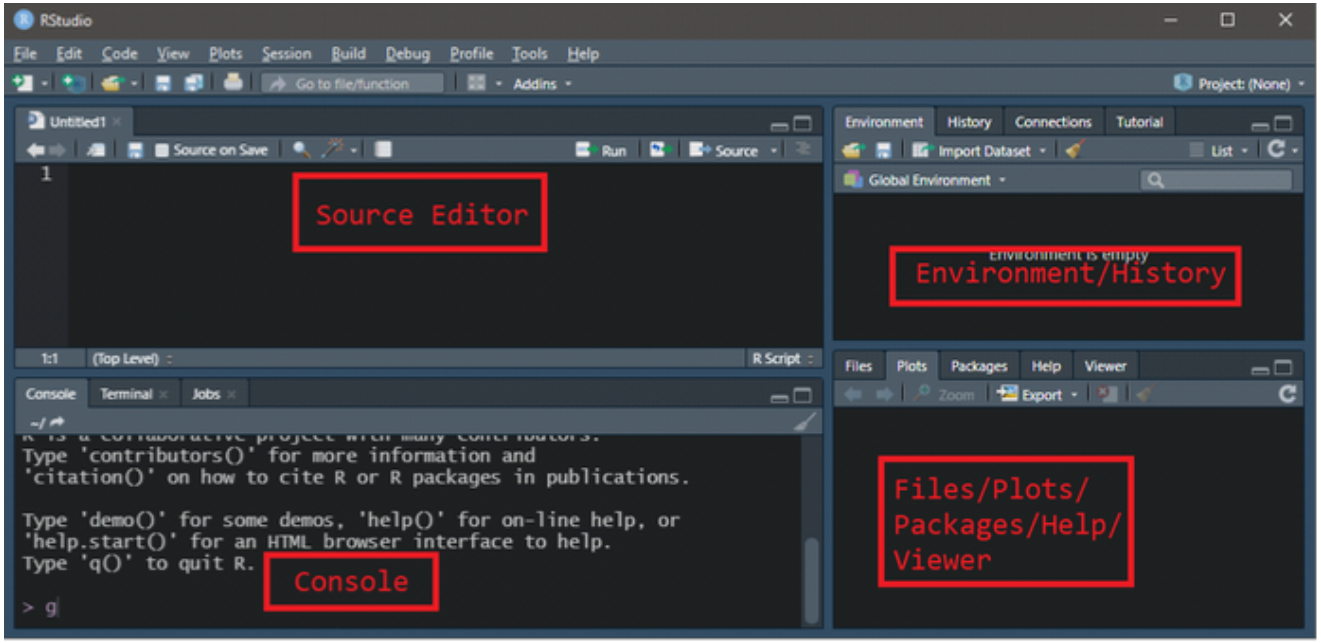
  residual e_i = observed y_i - predicted y_i

$$e_i = y_i - \hat{y}_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_p x_{ip})$$

- Question: is $e_i = \epsilon_i$?

In general $e_i \neq \epsilon_i$ since $\beta_j \neq \hat{\beta}_j$

RStudio Interface



- Type R codes in the [Source Editor], and hit [Ctrl+Enter] (Windows) or [Cmd+Enter] (Mac) to execute. Output in the [Console].
- Select several lines of R codes and hit [Ctrl+Enter] (or [Cmd+Enter] on Mac) to execute the selected lines.
- You can save your R codes in a .R file (R script) for reuse.

R as a calculator

```
3+2
```

```
## [1] 5
```

```
3*2
```

```
## [1] 6
```

```
3^2 # power
```

```
## [1] 9
```

```
log(100) # natural log
```

```
## [1] 4.60517
```

```
log10(100) # base 10 log
```

```
## [1] 2
```

R value assignment

We use assignment operator: `<-`

```
x <- 3+2   #Input
x
```

```
## [1] 5
```

```
x = log(100)
x
```

```
## [1] 4.60517
```

Vectors

The `c()` function can be used to create vectors of objects by concatenating things together.

c() stands for "combine" or "concatenate."

```
x <- c(0.5, 0.6)        ## numeric
x <- c(TRUE, FALSE)     ## logical
x <- c(T, F)            ## logical
x <- c("a", "b", "c")   ## character
x <- 9:29               ## integer
x <- c(1+0i, 2+4i)      ## complex
```

Other commonly used vectors

```
1:8
```

```
## [1] 1 2 3 4 5 6 7 8
```

```
seq(from = 3, to = 21, by =2)
```

```
##  [1]  3  5  7  9 11 13 15 17 19 21
```

```
rep(5,4)
```

```
## [1] 5 5 5 5
```

```
rep(c(1,2),4)
```

```
## [1] 1 2 1 2 1 2 1 2
```

Indexes of Vectors

```
z <- c(5,3,1,6,4,2)
z[3] # the 3rd element
```

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

```
z[c(1,4)] # the 1st and 4th elements [1] 5 6
```

```
## [1] 5 6
```

```
z[1:4] # the first 4 elements
```

```
## [1] 5 3 1 6
```

Negative indexes means excluding those elements

```
z[-1]
```

```
## [1] 3 1 6 4 2
```

Reading Data

There are a few principal functions reading data into R.

- `read.table`, `read.csv`, for reading tabular data
- `readLines`, for reading lines of a text file source, for reading in R code files
- `dget`, for reading in R code files
- `load`, for reading in saved workspaces.
- `unserialize`, for reading single R objects in binary form

For small to moderately sized datasets, you can usually call `read.table` without specifying any other arguments.

```
data <- read.table("foo.txt")
```

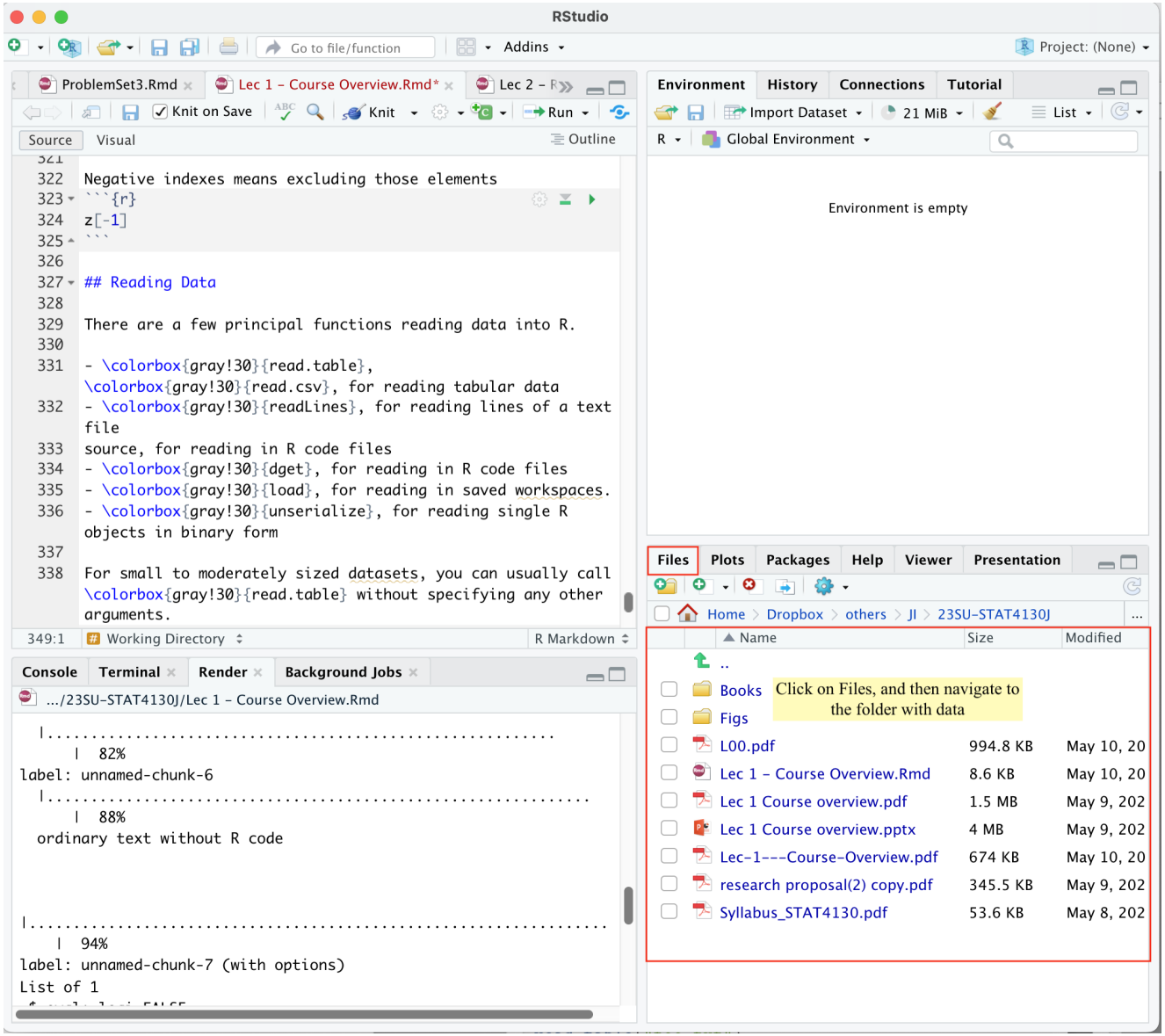
- R automatically figures out how many rows there are (and how much memory needs to be allocated)
- figure what type of variable is in each column of the table.

Working Directory

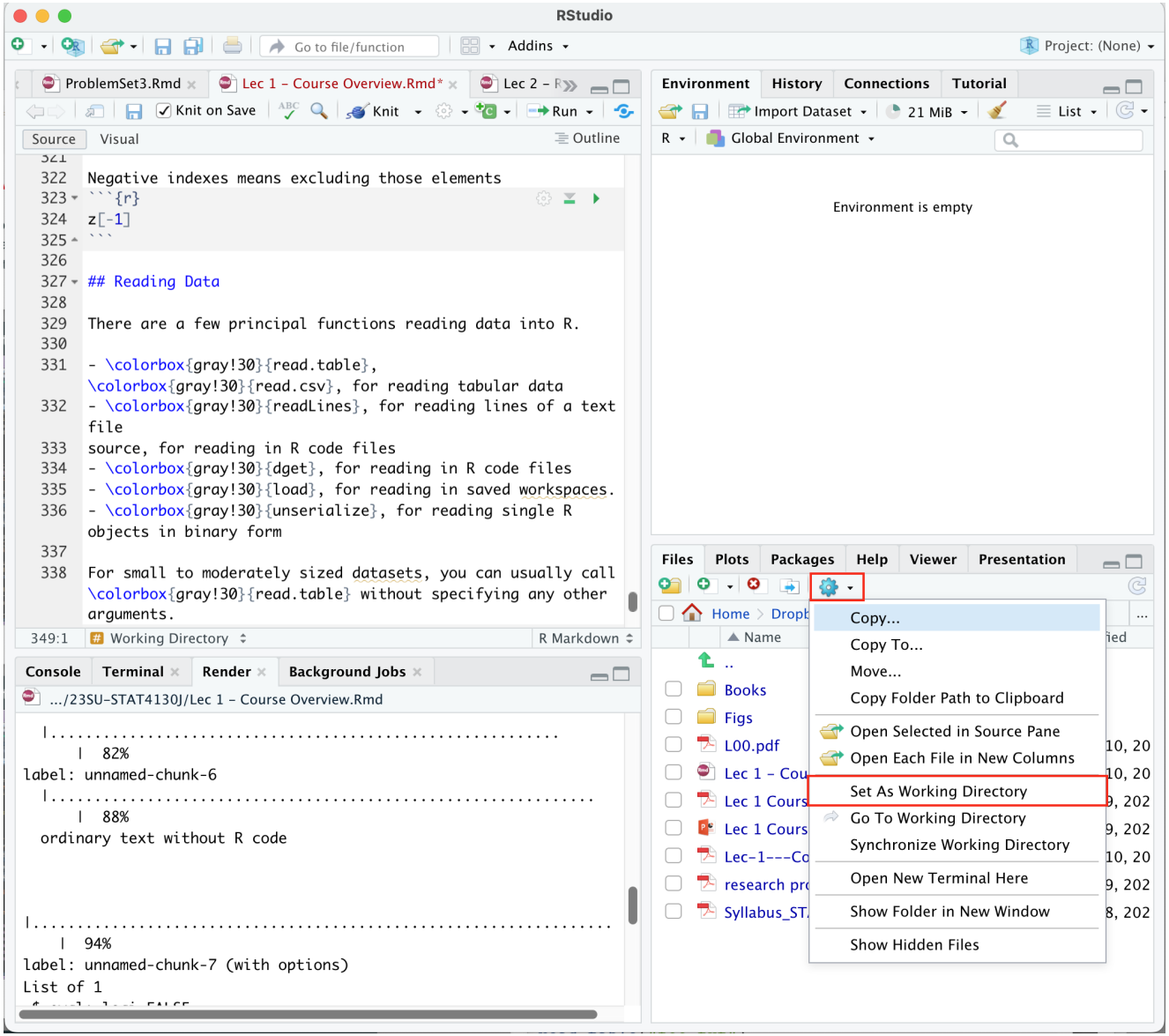
If we get an error message during reading that says “No such file or directory”.

This is because we haven’t told R where the file is located, which can be done by providing the complete path to the file or by setting the working directory to the folder the data file is located.

Set the Working Directory (1)



Set the Working Directory (2)



Demo: Childhood respiratory disease data

```
fevdata = read.table("fevdata.txt", header=TRUE)
dim(fevdata)
```

```
## [1] 654   5
```

We can see fevdata has 654 rows (cases) and 5 columns (variables).

Data: Smoking and FEV (Lung Capacity)

Sample of 654 youths, aged 3 to 19, in the area of East Boston during middle to late 1970's. The variables are

- age: Subject's age in years
- fev: Lung capacity of subject, measured by forced expiratory volume (abbreviated as FEV), the amount of air an individual can exhale in the first second of forceful breath in liters
- ht: Subject's height in inches
- sex: Gender of the subject coded as: 0 = Female, 1 = Male
- smoke:Smoking status coded as: 0 = Nonsmoker, 1 = Smoker

Summary Statistics

```
mean(fevdata$fev)
```

```
## [1] 2.63678
```

```
median(fevdata$fev)
```

```
## [1] 2.5475
```

```
sd(fevdata$fev)
```

```
## [1] 0.8670591
```

```
var(fevdata$fev)
```

```
## [1] 0.7517915
```

Data Summaries by Group using aggregate()

Mean FEV by Gender: (0 for females, 1 for males)

```
aggregate(fev ~ sex, data = fevdata, mean)
```

```
##   sex      fev
## 1   0 2.451170
## 2   1 2.812446
```

Better give meaningful labels for the categories rather than 0 and 1.

```
fevdata$sex = factor(fevdata$sex, labels=c("Female","Male"))
fevdata$smoke = factor(fevdata$smoke, labels=c("Nonsmoker","Smoker"))
aggregate(fev ~ sex, data = fevdata, mean)
```

```
##      sex      fev
## 1 Female 2.451170
## 2   Male 2.812446
```

```
aggregate(fev ~ smoke, data = fevdata, mean)
```

```
##       smoke      fev
## 1 Nonsmoker 2.566143
## 2    Smoker 3.276862
```

Summarizing Data By Two Grouping Variables

```
aggregate(fev ~ smoke + sex, data = fevdata, mean)
```

```
##       smoke    sex      fev
## 1 Nonsmoker Female 2.379211
## 2    Smoker Female 2.965949
## 3 Nonsmoker   Male 2.734381
## 4    Smoker   Male 3.743231
```

aggregate() + summary()

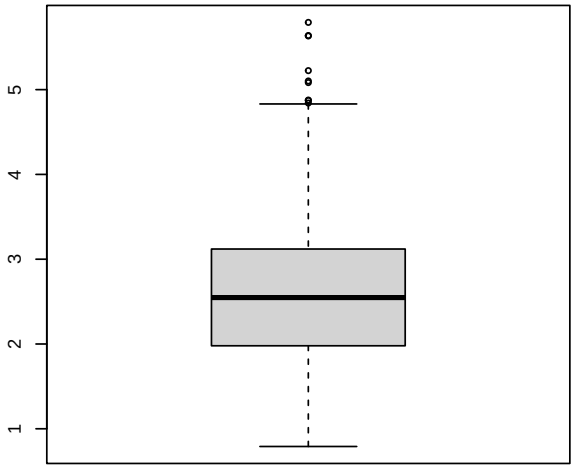
The summary() function can report more summary statistics.

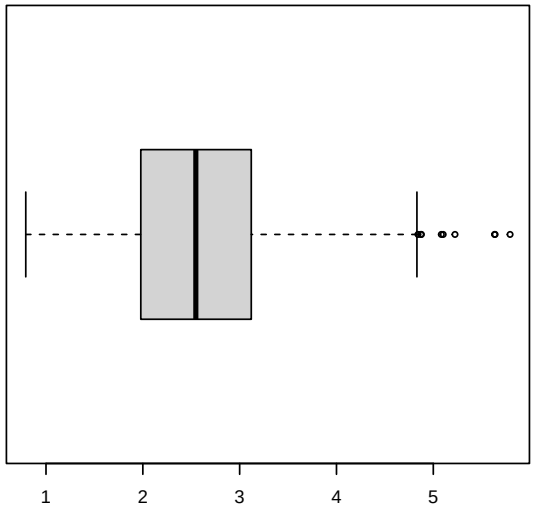
```
aggregate(fev ~ smoke + sex, data=fevdata, summary)
```

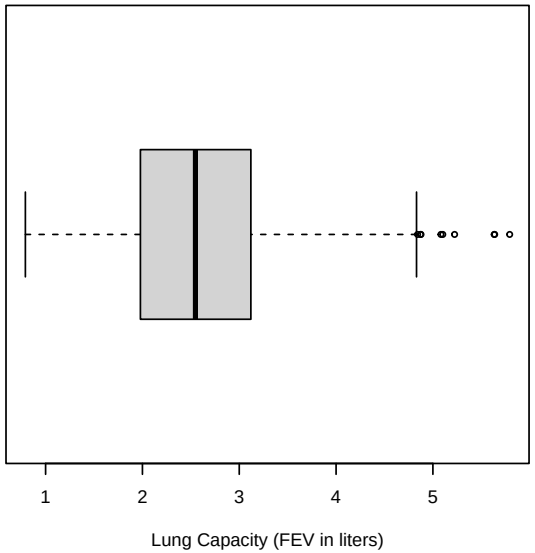
```
##       smoke    sex fev.Min. fev.1st Qu. fev.Median fev.Mean
## 1 Nonsmoker Female 0.791000    1.877500   2.417000 2.379211
## 2    Smoker Female 2.198000    2.678000   3.074000 2.965949
## 3 Nonsmoker   Male 0.796000    1.964250   2.547500 2.734381
## 4    Smoker   Male 1.694000    3.358750   3.878000 3.743231
##   fev.Max.
## 1 3.816000
## 2 3.835000
## 3 5.793000
## 4 4.872000
```

Visualizations - Boxplots

```
par(mfrow=c(1,3))
boxplot(fevdata$fev)
boxplot(fevdata$fev, horizontal = T)
boxplot(fevdata$fev, horizontal = T,
xlab="Lung Capacity (FEV in liters)")
```

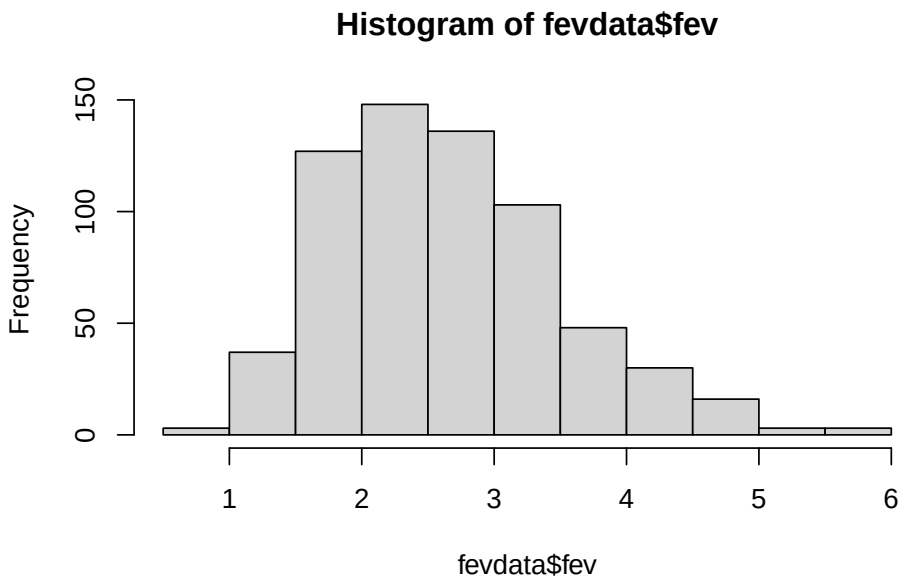


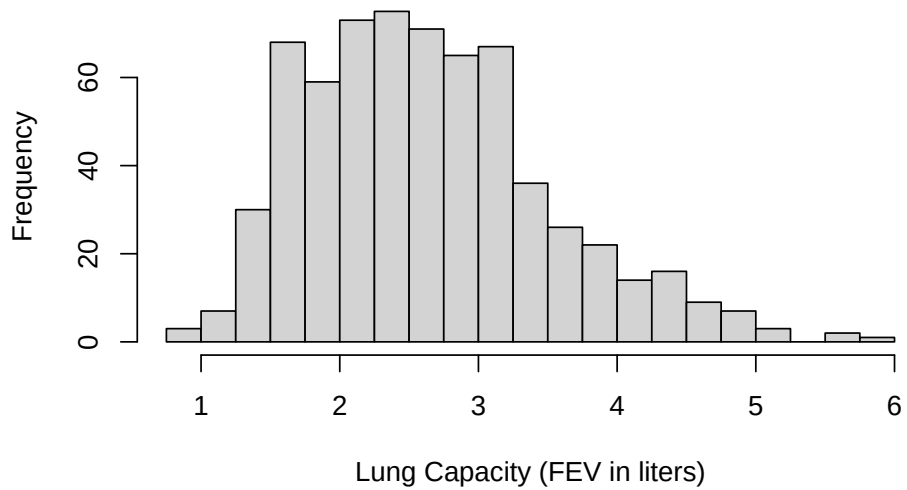




Visualization - Histogram

```
par(mfrow=c(1,2))
hist(fevdata$fev)
hist(fevdata$fev, breaks=seq(0.75,6, 0.25),
          xlab="Lung Capacity (FEV in liters)", main="")
```





Visualization - ggplot2

- "gg" in ggplot means = Grammer of Graphics
- a powerful and versatile toolkit for data visualization

```
library(ggplot2)
#If ggplot2 library is not installed. You can install it using
#install.packages("ggplot2")
```

In ggplot, aes() is the short hand for "aesthetic".

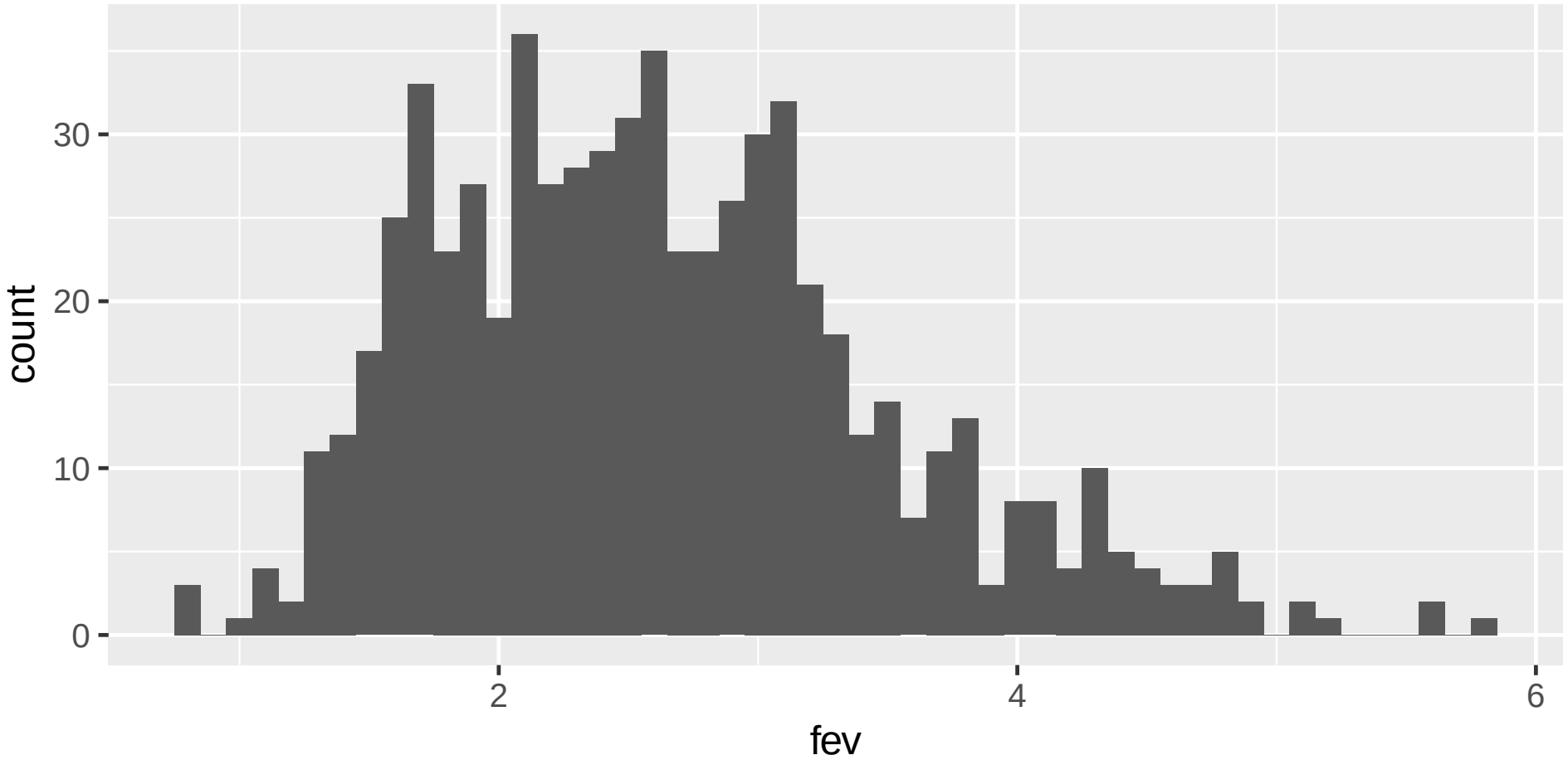
Variables involved in a ggplot must be enclosed within aes().

Introduction to R Graphics with ggplot2 (from Harvard)

https://iqss.github.io/dss-workshops/Rgraphics.html

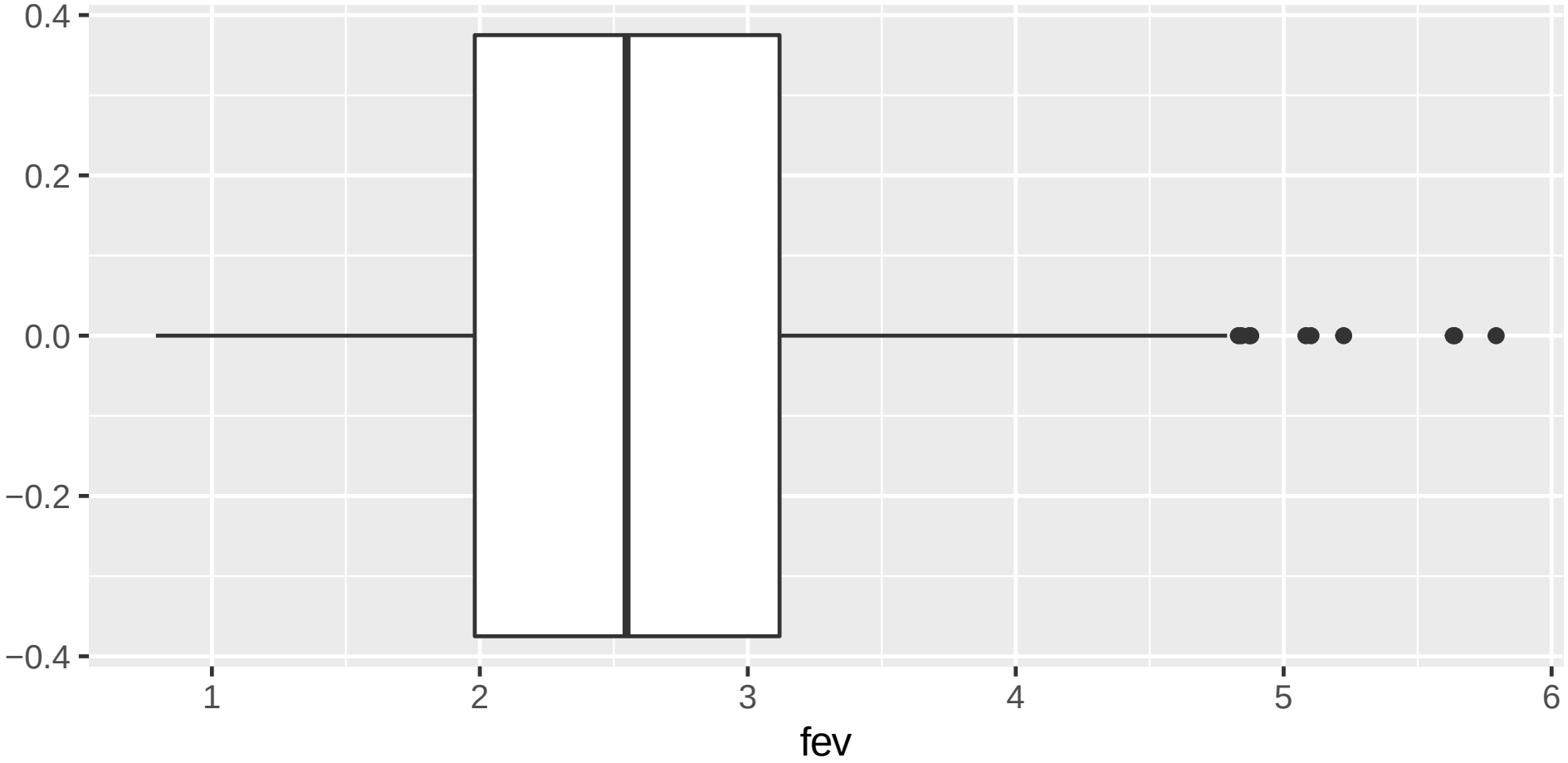
Histogram in ggplot2

```
ggplot(fevdata, aes(x = fev)) + geom_histogram(binwidth = 0.1)
```



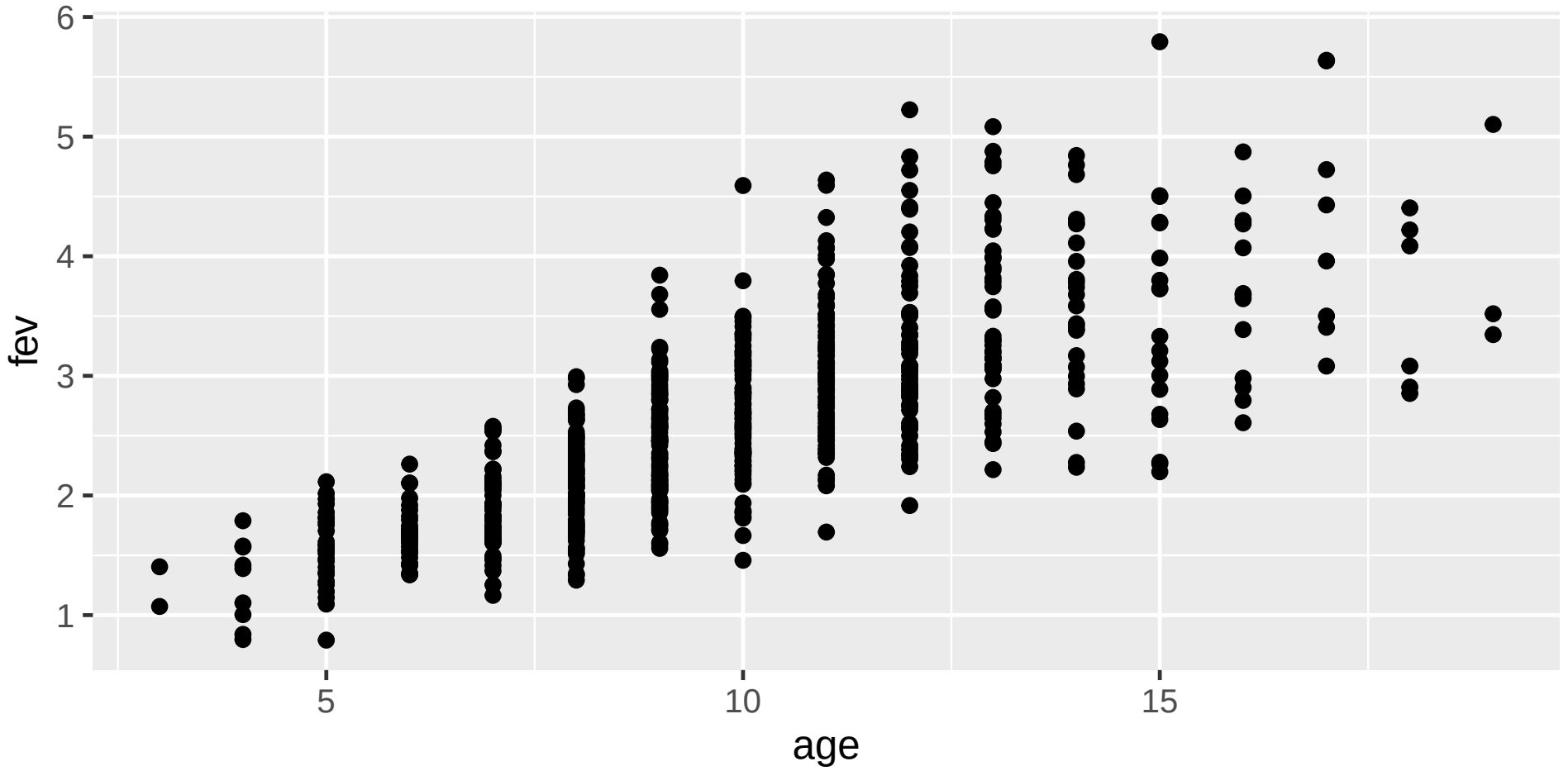
Boxplots in ggplot2

```
ggplot(fevdata, aes(x = fev)) + geom_boxplot()
```



Scatterplot in ggplot2

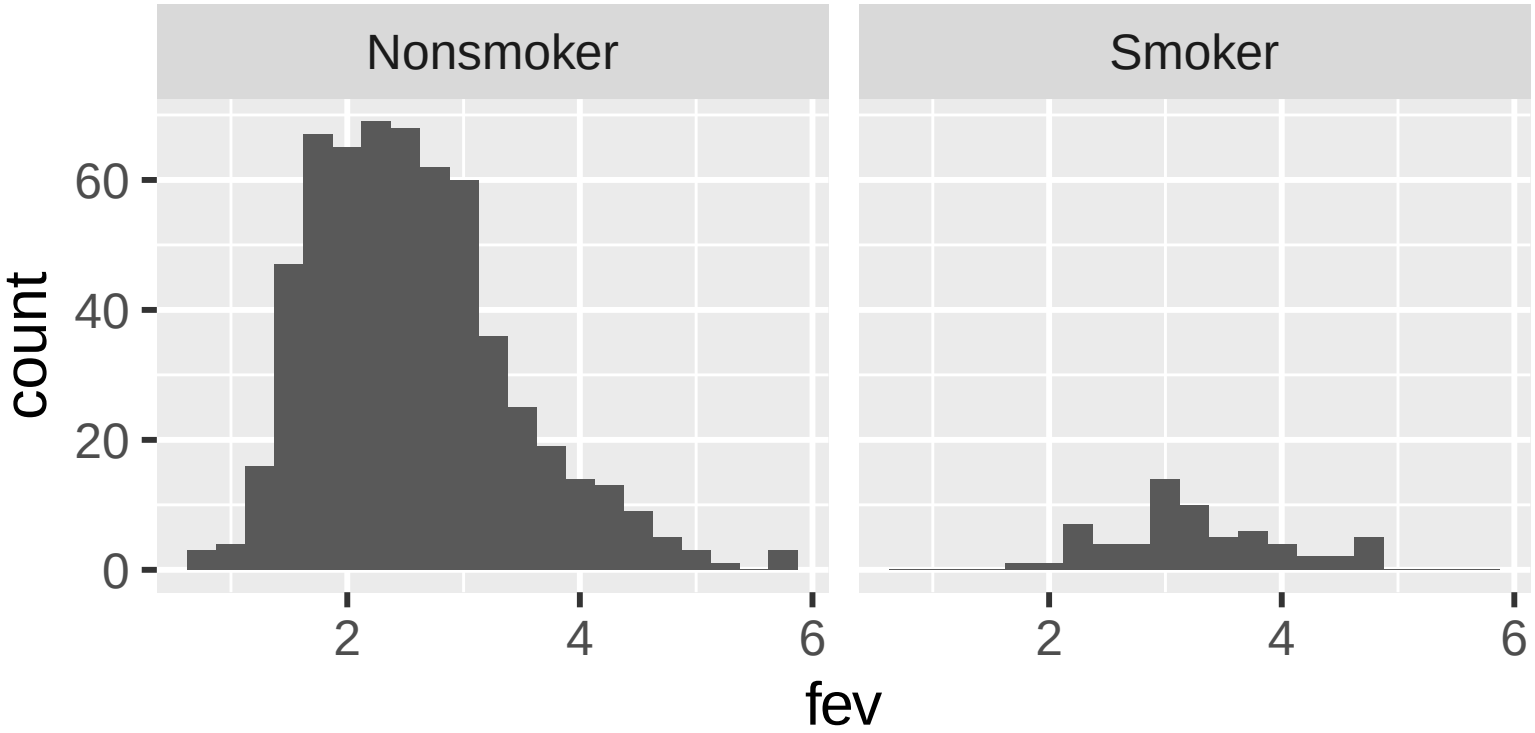
```
ggplot(fevdata, aes(x = age, y = fev)) + geom_point()
```



Facet — Inspect Data by Group

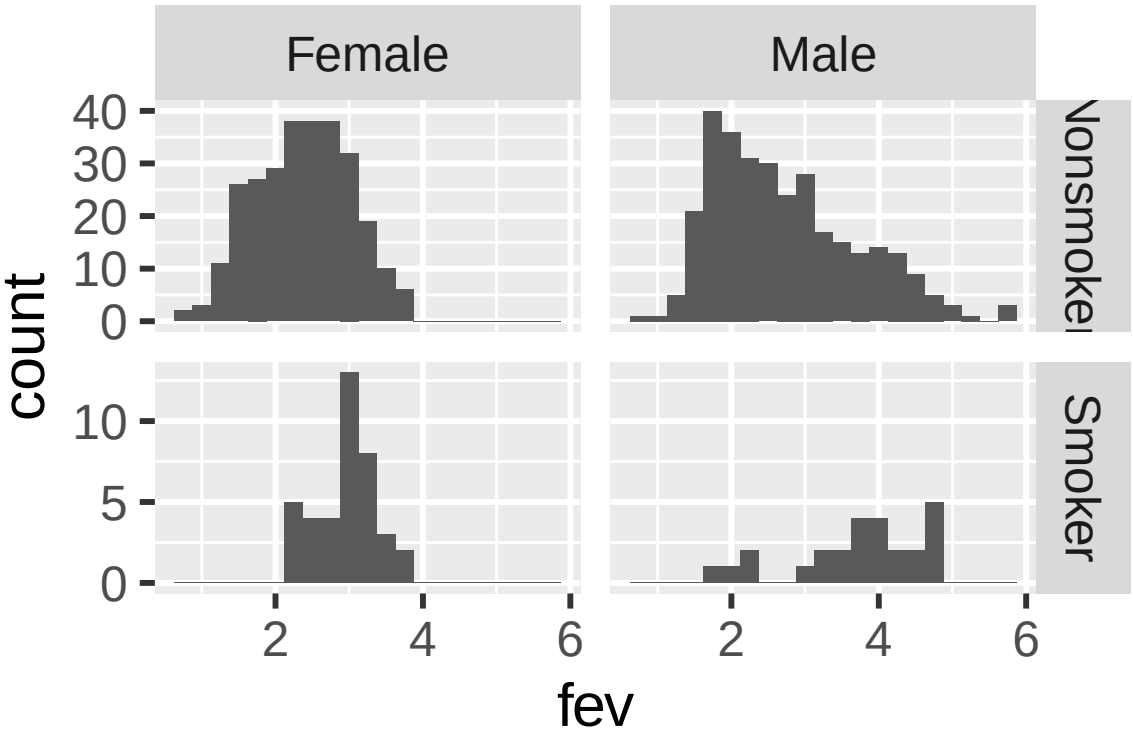
ggplot allows you to inspect data by group using facet_wrap.

```
ggplot(fevdata, aes(x = fev)) +
  geom_histogram(binwidth = 0.25) +
  facet_wrap(~smoke)
```



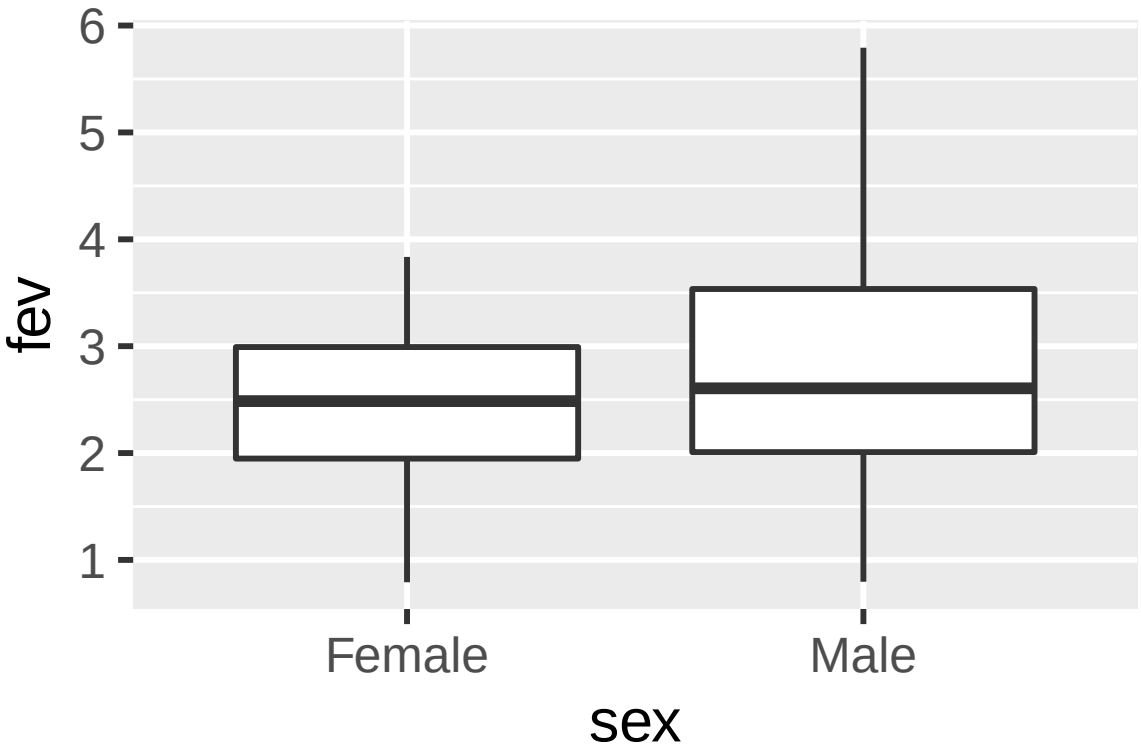
Facet Over 2 Grouping Varibles

```
ggplot(fevdata, aes(x = fev)) +
  geom_histogram(binwidth = 0.25) +
  facet_grid(smoke ~ sex, scale="free_y")
```



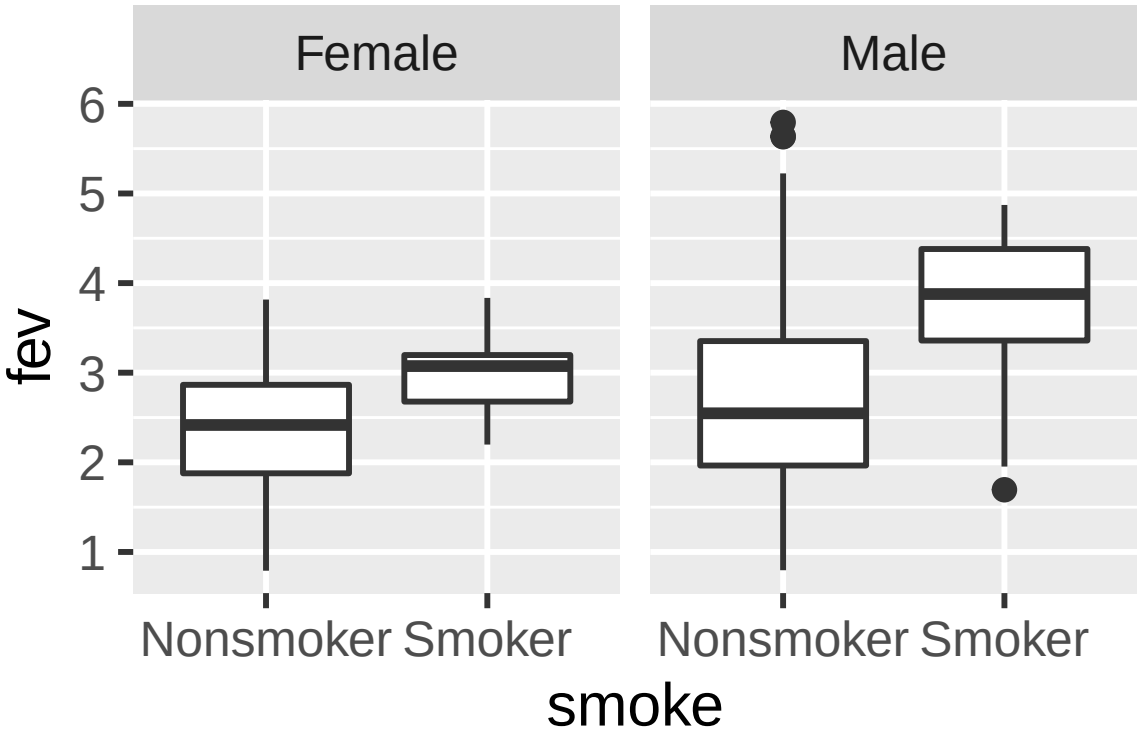
Boxplots by Group (or Side-by-Side Boxplots)

```
ggplot(fevdata, aes(x = sex, y = fev)) + geom_boxplot()
```



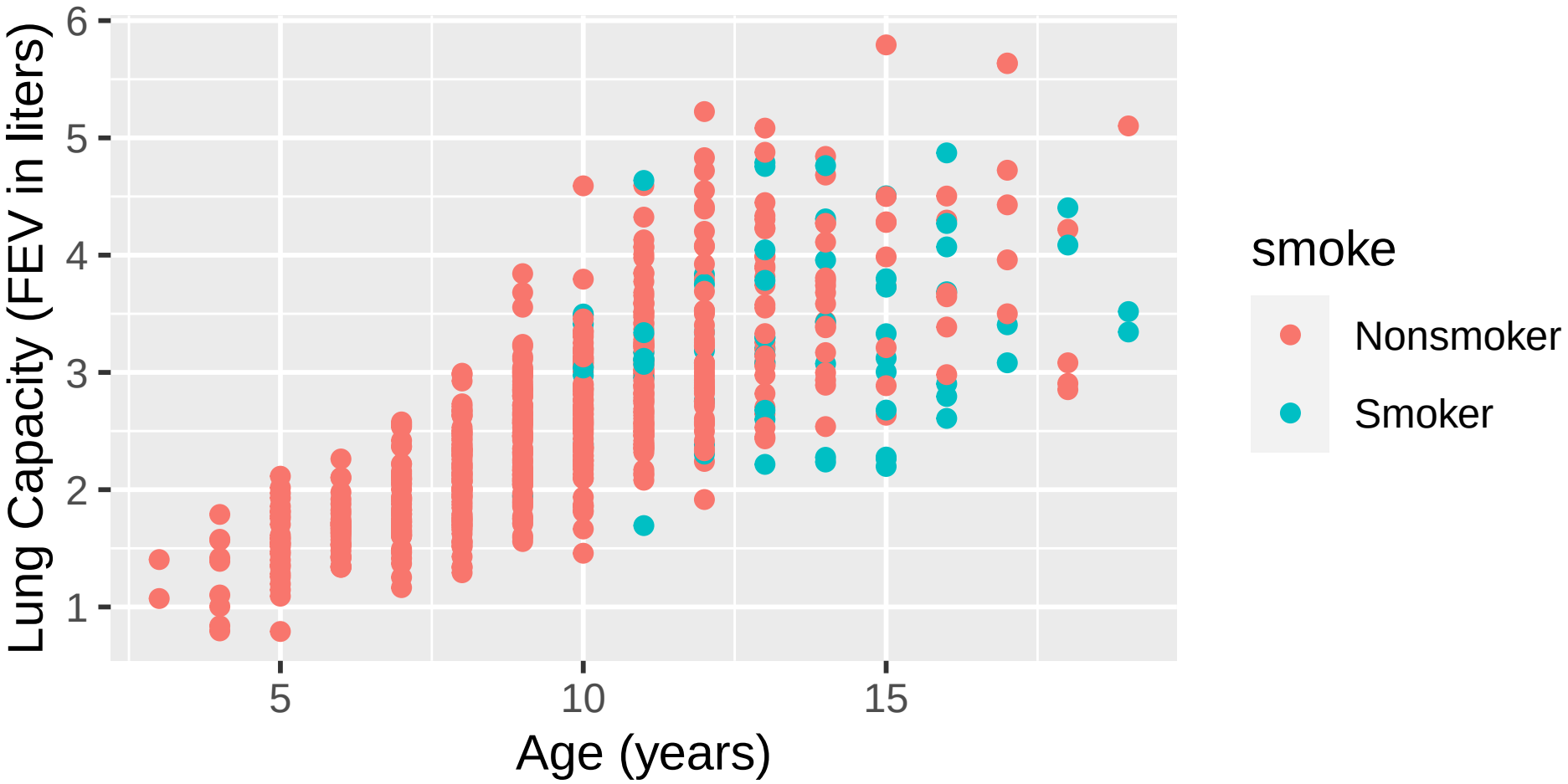
Faceting Boxplots

```
ggplot(fevdata, aes(x = smoke, y = fev)) +
geom_boxplot() +
facet_wrap(~sex)
```



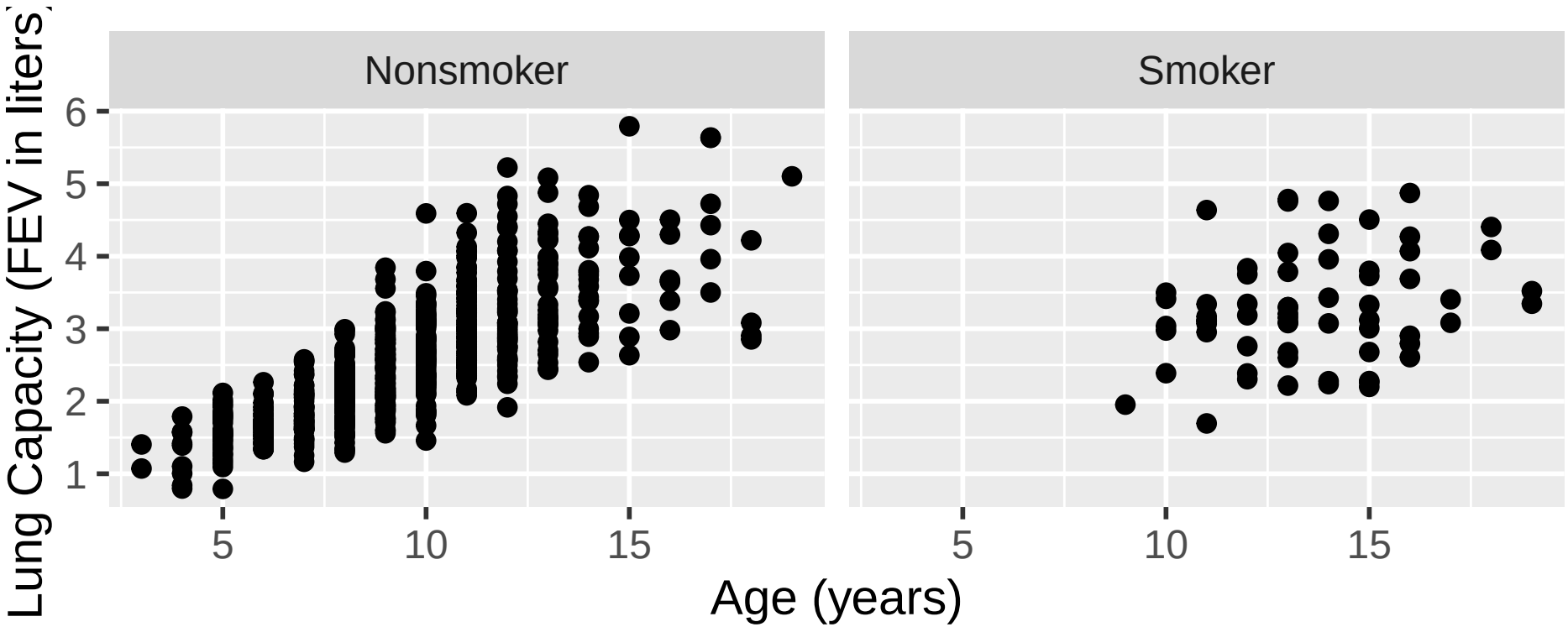
Color-coded Scatter Plots

```
ggplot(fevdata, aes(x = age, y = fev, col=smoke)) + geom_point() +
 xlab("Age (years)") + ylab("Lung Capacity (FEV in liters)")
```



Scatter Plots Faceted by Smoke Status

```
ggplot(fevdata, aes(x = age, y = fev)) +
geom_point() +
facet_wrap(~smoke) +
xlab("Age (years)") +
ylab("Lung Capacity (FEV in liters)")
```



facet_grid Over Both sex & smoke

```
ggplot(fevdata, aes(x = age, y = fev)) +
geom_point() +
facet_grid(smoke~sex) +
xlab("Age (years)") +
ylab("Lung Capacity (FEV in liters)")
```