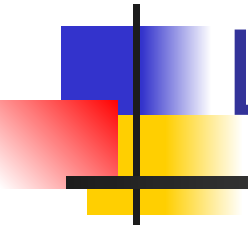
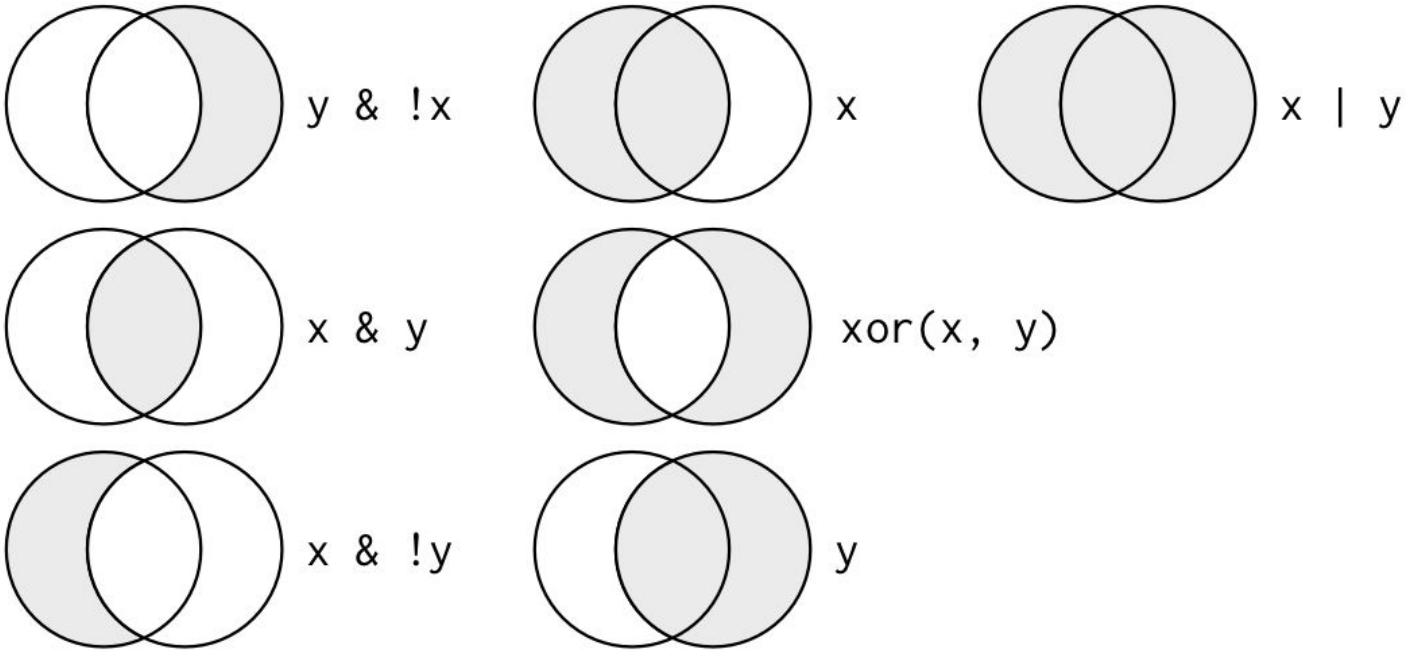


Lecture 7

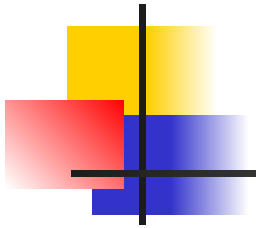
- Logical Operators in R
- Missing Data
- Recoding Missing Data
- Count of Missing Data

Set Operations



Logical Operators in R

Operators	Meaning
<	Less than
<=	Less than or equal to
>	More than
>=	More than or equal to
==	Equal to
!=	Not equal to
!a	Not a
a|b	a or b
a&b	a and b
isTRUE(a)	Test if a is true
%in%	in the set



Examples

- ## Finds all flights that departed in November or December:

```
filter(flights, month == 11 | month == 12)

nov_dec <- filter(flights, month %in% c(11, 12))
```

- ## Find flights that weren't delayed (on arrival or departure) by more than two hours:

```
filter(flights, !(arr_delay > 120 | dep_delay > 120))

filter(flights, arr_delay <= 120, dep_delay <= 120)
```

Missing Data

- Missing values represented by **NA** (not available)

  Impossible values (e.g., divide by zero) represented by **NaN** (not a number)

- Testing for missing values: is.na()

  - returns a logical vector with TRUE in the element locations that contain missing values represented by NA. is.na() will work on vectors, lists, matrices, and data frames.

```
> # vector with missing data
> x <- c(1:4, NA, 6:7, NA)
> x
[1]  1  2  3  4  NA  6  7  NA
>
> is.na(x)
[1] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE
>
> which(is.na(x))   ##identify locations in the vector with NAs
[1] 5 8
```

Recoding Missing Values

- Missing values can be recoded by
  - Recoding specific indicators that represent missing values.

```
># vector with missing data
>x <- c(1:4, NA, 6:7, NA)
>x[is.na(x)] <- mean(x, na.rm = TRUE)
#na.rm    is a logical parameter evaluating to TRUE or FALSE indicating whether NA
  values should be stripped before the computation proceeds.
>x
[1] 1.000000 2.000000 3.000000 4.000000 3.833333 6.000000 7.000000  3.833333
>round(x, 2)
[1] 1.00 2.00 3.00 4.00 3.83 6.00 7.00 3.83
```

  - Omitting all rows containing missing values.

```
> df<- data.frame(col1 = c(1:3, NA), col2 = c(2.5, 4.2, NA, 3.2))
> df
> na.omit(df)
```

```
  col1 col2
1    1  2.5
2    2  4.2
3    3   NA
4   NA  3.2
```

```
  col1 col2
1    1  2.5
2    2  4.2
```

Recoding Missing Values

- Missing values can be recoded by
  - Using normal sub-setting and assignment operations.

```
# data frame that codes missing values as 99
>df <- data.frame(col1 = c(1:3, 99), col2 = c(2.5, 4.2, 99, 3.2))
>df
```

```
  col1 col2
1    1  2.5
2    2  4.2
3    3 99.0
4   99  3.2
```

```
> # change 99s to NAs
>df[df == 99] <- NA
>df
```

```
  col1 col2
1    1  2.5
2    2  4.2
3    3   NA
4   NA  3.2
```

Count of Missing Data

- Missing values represented by **NA** (not available)

```
#create data frame
df <- data.frame(team=c('A', 'B', 'C', NA, 'E'),
                 points=c(99, 90, 86, 88, 95),
                 assists=c(NA, 28, NA, NA, 34),
                 rebounds=c(30, 28, 24, 24, NA))
df
```

	team	points	assists	rebounds
1	A	99	NA	30
2	B	90	28	28
3	C	86	NA	24
4	<NA>	88	NA	24
5	E	95	34	NA

- Location of missing values

```
> which(is.na(df$assists))
[1] 1 3 4
```

- Count number of missing values in a column

```
>sum(is.na(df$column_name))
```

```
> sum(is.na(df$team))
[1] 1
> sum(is.na(df$points))
[1] 0
> sum(is.na(df$assists))
[1] 3
> sum(is.na(df$rebounds))
[1] 1
```

- Total number of missing values in entire data

```
>sum(is.na(df))
```

```
> sum(is.na(df))
[1] 5
```