# Data Wrangling

Using the following functions: /filter/, /select/, /arrange/, /mutate/, /group by/ and /summarise/.

* Subset
* Reorder
* Add or modify columns
* Group summaries

All functions start with the dataset – in this case diamonds – i.e., filter(diamonds, xxxxx).

# Simple filter

### By Cut = Ideal

Diamonds\_sm = filter(diamonds, cut == ‘Ideal’) this filters diamonds by the cut ‘ideal’. The variable diamonds\_sm helps to visualise the result through the view function below;

Note: use diamonds\_sm <- instead of =. Though = works.

View(Diamonds\_sm)

### By price greater than 10,000

Diamonds\_sm <- filter(diamonds, price > 10000)

View(Diamonds\_sm)

### Multiple criteria

Diamonds\_sm <- filter(diamonds, price > 10000, cut == ‘Ideal’)

View(Diamonds\_sm)

NB: Do not have to type view() multiple times once the variable is updated.

OR = | (not exclusive).

* Filter for missing values use is.na (is missing) or !is.na (is not missing)

## SELECT()

Selects the data specified.

* Diamonds\_sm <- select(diamonds, cut, color)
* View(diamonds\_sm)

This command displays the cut and color columns in the diamonds database.

You can use numbers, i.e.

* diamonds\_sm <- select(diamonds, 1:4)

### Starts\_with()

* Diamonds\_sm <- select(diamonds, starts\_with(“c”))

### Ends\_with() & Contains()

* Diamonds\_sm <- select(diamonds, ends\_with(“c”))
* Diamonds\_sm <- select(diamonds, contains(“c”))

### Everything()

i.e. you want to put price before everything else

* Diamonds\_sm <- select(diamonds, price, everything())

### Eliminating a column: put a –

* Diamonds\_sm <- select(diamonds, -price)

# Reorder with arrange()

Arrange() arranges the table by a specified row;

* Diamonds\_arr <- diamonds %>%
  + - Arrange(color)

View(diamonds\_arr)

This shows the table arranged by color, descending order.

## Desc() and Asc()

* Diamonds\_arr <- diamonds %>%
  + - Arrange(asc(colour))

# Modify and add colums with mutate()

* diamonds\_new <- diamonds %>%

mutate(mass\_g = .20 \* carat)

glimpse(diamonds\_new)

view(diamonds\_new)

* diamonds\_new <- diamonds %>%

mutate(mass\_g = .20 \* carat, price\_per\_carat = price / carat)

glimpse(diamonds\_new)

* diamonds\_new <- diamonds %>%

mutate(mass\_g = .20 \* carat, price\_per\_carat = price / carat, cut =tolower(cut))

glimpse(diamonds\_new)

view(diamonds\_new)

### T/F

* diamonds\_new <- diamonds %>%
* mutate(mass\_g = .20 \* carat, price\_per\_carat = price / carat, cut =tolower(cut), expensive\_tf = price > 10000)
* glimpse(diamonds\_new)
* view(diamonds\_new)

Returns TRUE or FALSE if condition met

Further reading: slice, blind, joins, renames, case\_when

# Group Bys

* Diamonds %>%

Group\_by(cut) %>%

Summarise(mean(price))

### Can rename the summarised column

* Summarise(avg\_price = mean(price)

### Standard deviation

* Summarise(avg\_price = mean(price), sd\_price = sd(price))

n() – gives count, i.e. count = n())

* Can make up own group

i.e. group\_by(price > 10000)

### can rename the made up groupx

i.e. grou\_by(expensive = price > 10000)