**dplyr**

Functions

* select(): select variables
* filter(): filter by criteria
  + also see slice()
* group\_by(): groups by categorical levels
* arrange(): order data
* mutate():
  + also see transmute()
* summarise(): summary output
* sample\_n() and sample\_frac()
* join(): joining two dataframes (similar to joins in SQL)

library(readr)

library(dplyr)

soccerdata <- read\_csv("./data/soccer.csv")

dim(soccerdata)

head(soccerdata)

**select**

soccerdata %>%

select(type\_name, team\_name, now\_cost, total\_points)

**filter**

soccerdata %>%

select(type\_name, team\_name, now\_cost, total\_points) %>%

filter(now\_cost > 5 & total\_points > 30, team\_name == "Arsenal")

**group\_by and summarise**

soccerdata %>%

select(type\_name, team\_name, now\_cost, total\_points) %>%

group\_by(team\_name) %>%

summarise(teamcost = sum(now\_cost), teampoints = sum(total\_points))

soccerdata %>%

select(type\_name, team\_name, now\_cost, total\_points) %>%

group\_by(team\_name, type\_name) %>%

summarise(teamcost = sum(now\_cost), teampoints = sum(total\_points))

**arrange**

soccerdata %>%

select(type\_name, team\_name, now\_cost, total\_points) %>%

group\_by(team\_name, type\_name) %>%

summarise(teamcost = sum(now\_cost), teampoints = sum(total\_points)) %>%

arrange(desc(team\_name))

**mutate and transmute**

soccerdata %>%

select(type\_name, team\_name, now\_cost, total\_points) %>%

group\_by(team\_name) %>%

summarise(teamcost = sum(now\_cost), teampoints = sum(total\_points)) %>%

mutate(league.average = sum(teamcost)/n(),

cost\_diff = league.average - teamcost)

soccerdata %>%

select(type\_name, team\_name, now\_cost, total\_points) %>%

group\_by(team\_name) %>%

summarise(teamcost = sum(now\_cost), teampoints = sum(total\_points)) %>%

transmute(team\_name = team\_name,

league.average = sum(teamcost)/n(),

cost\_diff = league.average - teamcost)

**sample\_n() and sample\_frac()**

data.df <- data.frame(y1=rnorm(100),x1=rnorm(100),x2=rnorm(100))

head(data.df)

dim(data.df)

sample\_n(data.df, 70)

sample\_frac(data.df, .6)

**joins**

1. inner\_join(x, y)

* all rows from x where there are matching values in y
* **ALL** columns from x **AND** y
* if there are multiple matches between x and y
  + all combination of the matches are returned

library(readr)

flavors <- read\_csv("./data/icecream\_flavors.csv")

flavors

brands <- read\_csv("./data/icecream\_brands.csv")

brands

inner\_join(flavors, brands)

1. semi\_join(x, y)

* all rows from x where there are matching values in y
* *only columns from x*
* *won't return duplicate rows*

semi\_join(flavors, brands)

1. left\_join(x, y)

* all rows from x
* **ALL** columns from x **AND** y
* *all combination of matches*

left\_join(flavors, brands)

1. anti\_join(x, y)

* all rows from x where **NO** matching values in y
* only columns from x

anti\_join(flavors, brands)