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)
```