# Tidyverse Demo

Demonstrating tidyverse packages and functions.

### Create a tibble

```
# use a mlbench data frame
library(mlbench)
data("PimaIndiansDiabetes2")

library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.0     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   0.8.5
## v tidyr   1.0.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
tb <- tbl_df(PimaIndiansDiabetes2)
tb
```

```
## # A tibble: 768 x 9
##    pregnant glucose pressure triceps insulin  mass pedigree   age diabetes
##       <dbl>   <dbl>    <dbl>   <dbl>   <dbl> <dbl>    <dbl> <dbl> <fct>
##  1        6     148       72      35      NA  33.6    0.627    50 pos
##  2        1      85       66      29      NA  26.6    0.351    31 neg
##  3        8     183       64      NA      NA  23.3    0.672    32 pos
##  4        1      89       66      23      94  28.1    0.167    21 neg
##  5        0     137       40      35     168  43.1    2.29     33 pos
##  6        5     116       74      NA      NA  25.6    0.201    30 neg
##  7        3      78       50      32      88  31      0.248    26 pos
##  8       10     115       NA      NA      NA  35.3    0.134    29 neg
##  9        2     197       70      45     543  30.5    0.158    53 pos
## 10        8     125       96      NA      NA  NA      0.232    54 pos
## # ... with 758 more rows
```

```
# remove the data frame to free up memory
rm(PimaIndiansDiabetes2)
```

A glimpse is a view similar to str.

```
glimpse(tb)
```

```
## Rows: 768
## Columns: 9
## $ pregnant <dbl> 6, 1, 8, 1, 0, 5, 3, 10, 2, 8, 4, 10, 10, 1, 5, 7, 0, 7, 1...
## $ glucose  <dbl> 148, 85, 183, 89, 137, 116, 78, 115, 197, 125, 110, 168, 1...
```

```
## $ pressure <dbl> 72, 66, 64, 66, 40, 74, 50, NA, 70, 96, 92, 74, 80, 60, 72...
## $ triceps  <dbl> 35, 29, NA, 23, 35, NA, 32, NA, 45, NA, NA, NA, NA, 23, 19...
## $ insulin  <dbl> NA, NA, NA, 94, 168, NA, 88, NA, 543, NA, NA, NA, NA, 846,...
## $ mass     <dbl> 33.6, 26.6, 23.3, 28.1, 43.1, 25.6, 31.0, 35.3, 30.5, NA, ...
## $ pedigree <dbl> 0.627, 0.351, 0.672, 0.167, 2.288, 0.201, 0.248, 0.134, 0....
## $ age      <dbl> 50, 31, 32, 21, 33, 30, 26, 29, 53, 54, 30, 34, 57, 59, 51...
## $ diabetes <fct> pos, neg, pos, neg, pos, neg, pos, neg, pos, pos, neg, pos...
```

### The dplyr package

Some dply functions work on columns. These are demonstrated below.

### select()

Select a subset of columns. The select() function returns a tibble but it was not saved and will be discarded after the glimpse is output.

```
select(tb, diabetes, pregnant) %>%
  glimpse
```

```
## Rows: 768
## Columns: 2
## $ diabetes <fct> pos, neg, pos, neg, pos, neg, pos, neg, pos, pos, neg, pos...
## $ pregnant <dbl> 6, 1, 8, 1, 0, 5, 3, 10, 2, 8, 4, 10, 10, 1, 5, 7, 0, 7, 1...
```

or:

```
tb %>%
  select(diabetes, pregnant) %>%
  glimpse
```

```
## Rows: 768
## Columns: 2
## $ diabetes <fct> pos, neg, pos, neg, pos, neg, pos, neg, pos, pos, neg, pos...
## $ pregnant <dbl> 6, 1, 8, 1, 0, 5, 3, 10, 2, 8, 4, 10, 10, 1, 5, 7, 0, 7, 1...
```

### mutate()

The mutate() function can create new columns from old ones.

```
tb <- tb %>%
       mutate(glucose_high = factor(
         ifelse(glucose>mean(glucose, na.rm=TRUE), 1, 0)))

tb[1:5, c(2, 10)]
```

```
## # A tibble: 5 x 2
##   glucose glucose_high
##     <dbl> <fct>
## 1     148 1
## 2      85 0
## 3     183 1
## 4      89 0
## 5     137 1
```

We can also use mutate to delete a column by setting it to NULL.

```
tb <- tb %>%
  mutate(glucose_high = NULL)
```

```
names(tb)
```

```
## [1] "pregnant" "glucose"  "pressure" "triceps"  "insulin"  "mass"     "pedigree"
## [8] "age"      "diabetes"
```

### rename()

Rename a column.

```
tb <- rename(tb, blood_pressure = pressure)
```

### filter()

The filter function can select rows.

```
tb <- filter(tb, !is.na(glucose), !is.na(mass))
glimpse(tb)
```

```
## Rows: 752
## Columns: 9
## $ pregnant       <dbl> 6, 1, 8, 1, 0, 5, 3, 10, 2, 4, 10, 10, 1, 5, 7, 0, 7...
## $ glucose        <dbl> 148, 85, 183, 89, 137, 116, 78, 115, 197, 110, 168, ...
## $ blood_pressure <dbl> 72, 66, 64, 66, 40, 74, 50, NA, 70, 92, 74, 80, 60, ...
## $ triceps        <dbl> 35, 29, NA, 23, 35, NA, 32, NA, 45, NA, NA, NA, 23, ...
## $ insulin        <dbl> NA, NA, NA, 94, 168, NA, 88, NA, 543, NA, NA, NA, 84...
## $ mass           <dbl> 33.6, 26.6, 23.3, 28.1, 43.1, 25.6, 31.0, 35.3, 30.5...
## $ pedigree       <dbl> 0.627, 0.351, 0.672, 0.167, 2.288, 0.201, 0.248, 0.1...
## $ age            <dbl> 50, 31, 32, 21, 33, 30, 26, 29, 53, 30, 34, 57, 59, ...
## $ diabetes       <fct> pos, neg, pos, neg, pos, neg, pos, neg, pos, neg, po...
```

### arrange()

The following code arranges the rows by mass in descending order.

```
arrange(tb, desc(mass))
```

```
## # A tibble: 752 x 9
##    pregnant glucose blood_pressure triceps insulin  mass pedigree   age diabetes
##       <dbl>   <dbl>          <dbl>   <dbl>   <dbl> <dbl>    <dbl> <dbl> <fct>
##  1        0     129            110      46     130  67.1    0.319    26 pos
##  2        0     180             78      63      14  59.4    2.42     25 pos
##  3        3     123            100      35     240  57.3    0.88     22 neg
##  4        1      88             30      42      99  55      0.496    26 pos
##  5        0     162             76      56     100  53.2    0.759    25 pos
##  6        5     115             98      NA      NA  52.9    0.209    28 pos
##  7       11     135             NA      NA      NA  52.3    0.578    40 pos
##  8        0     165             90      33     680  52.3    0.427    23 neg
##  9        7     152             88      44      NA  50      0.337    36 pos
## 10        1     122             90      51     220  49.7    0.325    31 pos
## # ... with 742 more rows
```

### summarize

The summarize function computes statistical summaries of the data.

```
tb %>%
  summarize(min=min(mass), max=max(mass), sd=sd(mass))
```

```
## # A tibble: 1 x 3
##     min   max    sd
##   <dbl> <dbl> <dbl>
## 1  18.2  67.1  6.93
```

Another example:

```
tb %>%
  summarize(num_diabetic = sum(diabetes=="pos"), num_healthy = sum(diabetes=="neg"))
```

```
## # A tibble: 1 x 2
##   num_diabetic num_healthy
##          <int>       <int>
## 1          264         488
```

**group_by**

```
tb %>%
  group_by(diabetes) %>%
  summarize(median_BMI = median(mass, na.rm=TRUE))
```

```
## # A tibble: 2 x 2
##   diabetes median_BMI
##   <fct>         <dbl>
## 1 neg            30.1
## 2 pos            34.2
```