dplyr and Basic Statistics with R

Working with Rows / Cases

```
df1 <- filter(tbl, col1>10 & col2=="Yes")   Logical row filter with AND condition
df2 <- filter(tbl, col1>10 | col2=="Yes")   Logical row filter with OR condition
df3 <- distinct(tbl, col1)  Remove rows with duplicate values
df4 <- sample_frac(tbl, 0.n)    Sample of given % of rows (n = fraction)
df5 <- sample_n(tbl, N) Sample given 'N' number of rows
df6 <- slice(tbl, 6:10)     Selects rows using index
df7 <- top_n(tbl, n, col1)  Selects top-n entries based on col1
df8 <- arrange(tbl, col1)   Sort rows based on col1 in ascending order
df9 <- arrange(tbl, desc(col1)) Sort rows based on col1 in descending order
grouped_df <- group_by(tbl, col1)   Copy of data grouped by col1
ungrouped_df <- ungroup(grouped_df) Copy of ungrouped data
```

Working with Columns / Variables

```
df1 <- select(tbl, col1, col2)  Select columns from data explicitly
df2 <- select(tbl, starts_with("wildcard"), col2, matches("wildcard"))  Select columns
from data using helpers
df3 <- rename(tbl, col1_new = col1_old)     Rename columns
df4 <- mutate(tbl, col_new = col1*100)  Add new computed column
df5 <- transmute(tbl, col_new = col1_old*100)   Add new computed column & drop old
column
df6 <- mutate_at(tbl, vars(-col1), funs(log(.)))    Apply function on a specific column
```

Chaining Using Pipes

```
df1 <- df %>% select(col1, col2, col3) %>% filter(col1>10 & col2=="Yes")
Combine multiple operations in a natural order by using %>%
```

statinfer.com **Training and R&D - Data Science and Deep Learning**

Combine Tables Using Join

```
innerjoin_df <- inner_join(tbl1, tbl2) Mutating Inner Join
Return all rows from tbl1 where there are matching values in tbl2, and all columns from tbl1 and tbl2. If there are multiple matches between tbl1 and tbl2, all
combination of the matches is returned.
semijoin_df <- semi_join(tbl1, tbl2) Filtering Semi Join
Return all rows from tbl1 where there are matching values in tbl2, keeping just columns from tbl1. A semi join differs from an inner join because an inner join will return
one row of tbl1 for each matching row of tbl2, where a semi join will never duplicate rows of tbl1
leftjoin_df <- left_join(tbl1, tbl2) Mutating Left Join
Return all rows from tbl1, and all columns from tbl1 and tbl2. If there are multiple matches between tbl1 and tbl2, all combination of the matches is returned.
antijoin_df <- anti_join(tbl1, tbl2) Filtering Anti Join
Return all rows from tbl1 where there are not matching values in tbl2, keeping just columns from tbl1.
fulljoin_df <- full_join(tbl1, tbl2) Mutating Full Join
Return all rows and all columns from both tbl1 and tbl2. Where there are not matching values, returns NA for the one missing.
```

Basic Statistics

```
sample1_df <- df[sample(1:nrow(df), n), ]  n - sample size
sample2_df <- df[sample(1:nrow(df), n, replace=F, set.seed(S)) S - seed
mean(df$col_name1) Mean
var(df$col_name1)  Variance
median(df$col_name1)   Median
sd(df$col_name1)   Standard Deviation
quantile(df $col_name1, seq(0, 1, by=0.1)) Deciles
quantile(df$col_name1, seq(0, 1, by=0.01))  Percentiles
```

Graphs

```
boxplot(df_name$col_name)  Box Plot
plot(df_name$col_name)  Trend Plot
plot(df_name$col_name1, df_name$col_name2)  Scatter Plot
barplot(df_name, main="Title", xlab="col_name1", ylab="col_name2", col=4) Bar Plot
```