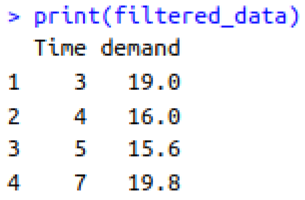
1. filter()

Code:

```
library(dplyr)
filtered_data <- BOD %>%
  filter(demand > 15)
print(filtered_data)
```

Output:



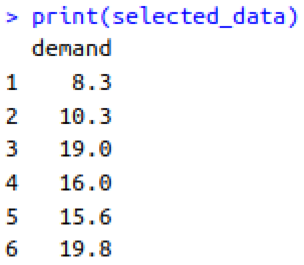
```
> print(filtered_data)
  Time demand
1    3   19.0
2    4   16.0
3    5   15.6
4    7   19.8
```

2. select()

Code:

```
library(dplyr)
selected_data <- BOD %>%
  select(demand)
print(selected_data)
```

Output:



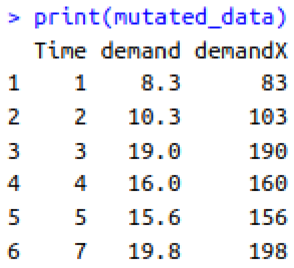
```
> print(selected_data)
  demand
1    8.3
2   10.3
3   19.0
4   16.0
5   15.6
6   19.8
```

3. mutate()

Code:

```
library(dplyr)
mutated_data <- BOD %>%
  mutate(demandX=demand*10)
print(mutated_data)
```

Output:



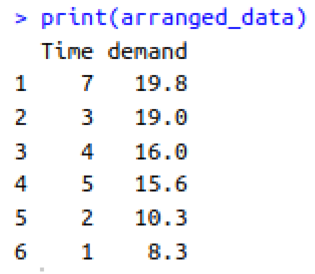
```
> print(mutated_data)
  Time demand demandX
1    1    8.3      83
2    2   10.3     103
3    3   19.0     190
4    4   16.0     160
5    5   15.6     156
6    7   19.8     198
```

4. arrange()

Code:

```
library(dplyr)
arranged_data <- BOD %>%
  arrange(desc(demand))
print(arranged_data)
```

Output:



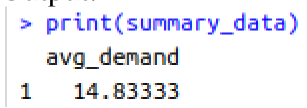
```
> print(arranged_data)
  Time demand
1    7   19.8
2    3   19.0
3    4   16.0
4    5   15.6
5    2   10.3
6    1    8.3
```

5. summarize()

Code:

```
library(dplyr)
summary_data <- BOD %>%
  summarize(avg_demand= mean(demand))
print(summary_data)
```

Output:



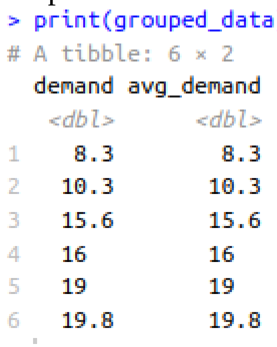
```
> print(summary_data)
  avg_demand
1   14.83333
```

6. group_by

Code:

```
library(dplyr)
grouped_data <- BOD %>%
  group_by(demand) %>%
  summarize(avg_demand= mean(demand))
print(grouped_data)
```

Output:



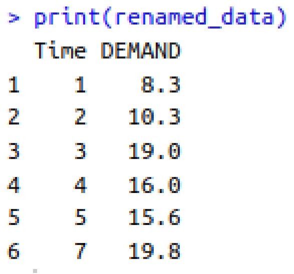
```
> print(grouped_data)
# A tibble: 6 × 2
  demand avg_demand
   <dbl>      <dbl>
1    8.3        8.3
2   10.3       10.3
3   15.6       15.6
4   16         16  
5   19         19  
6   19.8       19.8
```

7. rename()

Code:

```
library(dplyr)
renamed_data <- BOD %>%
  rename(DEMAND =demand)
print(renamed_data)
```

Output:



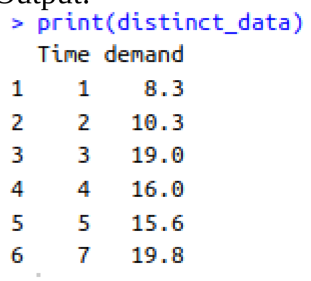
```
> print(renamed_data)
  Time DEMAND
1    1    8.3
2    2   10.3
3    3   19.0
4    4   16.0
5    5   15.6
6    7   19.8
```

8. distinct

Code:

```
library(dplyr)
distinct_data <- BOD %>%
  distinct(Time, demand)
print(distinct_data)
```

Output:



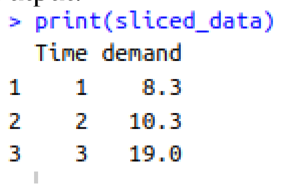
```
> print(distinct_data)
  Time demand
1    1    8.3
2    2   10.3
3    3   19.0
4    4   16.0
5    5   15.6
6    7   19.8
```

9. slice

Code:

```
library(dplyr)
sliced_data <- BOD %>%
  slice(1:3)
print(sliced_data)
```

Output:



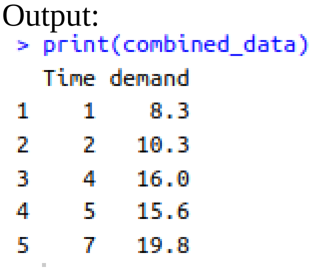
```
> print(sliced_data)
  Time demand
1    1    8.3
2    2   10.3
3    3   19.0
```

10. bind_rows() and bind_cols()

Code:

```
data1 <- BOD[1:2, ]
data2 <- BOD[4:6, ]
combined_data <- bind_rows(data1, data2)
print(combined_data)
```

Output:



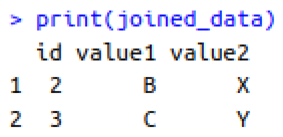
```
> print(combined_data)
  Time demand
1    1    8.3
2    2   10.3
3    4   16.0
4    5   15.6
5    7   19.8
```

11. joins

Code:

```
data1 <- data.frame(id = 1:3, value1 = c("A", "B", "C"))
data2 <- data.frame(id = 2:4, value2 = c("X", "Y", "Z"))
joined_data <- inner_join(data1, data2, by = "id")
print(joined_data)
```

Output:



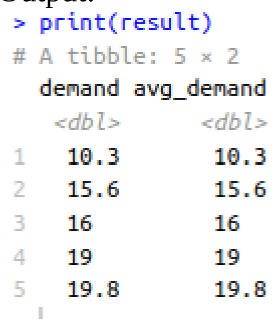
```
> print(joined_data)
  id value1 value2
1  2      B      X
2  3      C      Y
```

12. pipe

Code:

```
result <- BOD %>%
  filter(demand > 10) %>%
  group_by(demand) %>%
  summarize(avg_demand = mean(demand))
print(result)
```

Output:



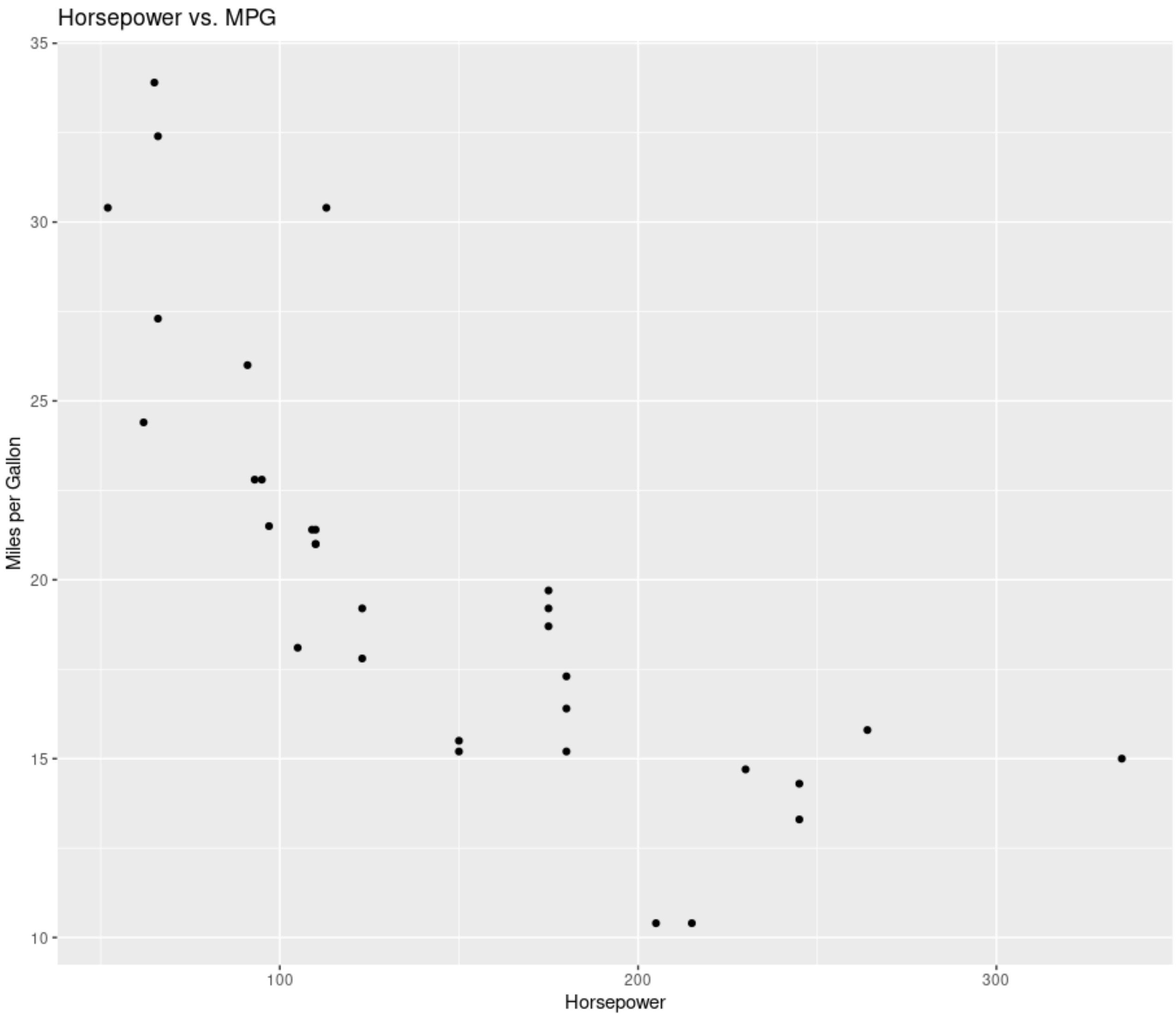
```
> print(result)
# A tibble: 5 × 2
  demand avg_demand
   <dbl>      <dbl>
1   10.3       10.3
2   15.6       15.6
3   16         16  
4   19         19  
5   19.8       19.8
```

2. ggplot2

Code:

```
library(ggplot2)
# Scatter plot of mpg vs. hp
ggplot(mtcars, aes(x = hp, y = mpg)) + geom_point() +labs(title = "Horsepower vs. MPG", x = 
"Horsepower", y = "Miles per Gallon")
```

Output:

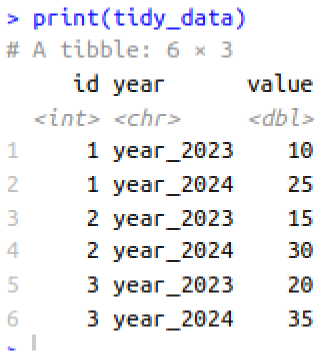


3. tidyr

Code:

```
library(tidyr)
# Sample messy dataset
data <- data.frame(
  id = 1:3,
  year_2023 = c(10, 15, 20),
  year_2024 = c(25, 30, 35)
)
# Transform data to a tidy format
tidy_data <- data %>%
  pivot_longer(cols = starts_with("year"), names_to = "year",
               values_to = "value")
print(tidy_data)
```

Output:



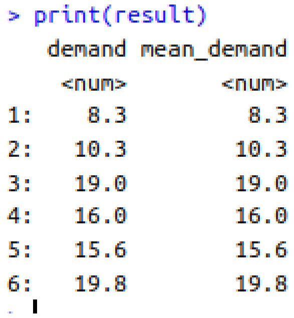
```
> print(tidy_data)
# A tibble: 6 × 3
     id year      value
  <int> <chr>     <dbl>
1     1 year_2023    10
2     1 year_2024    25
3     2 year_2023    15
4     2 year_2024    30
5     3 year_2023    20
6     3 year_2024    35
```

4. data.table

Code:

```
library(data.table)
# Create a data.table
data <- data.table(mtcars)
# Calculate mean mpg by number of cylinders
result <- data[, .(mean_mpg = mean(mpg)), by = cyl]
print(result)
```

Output:



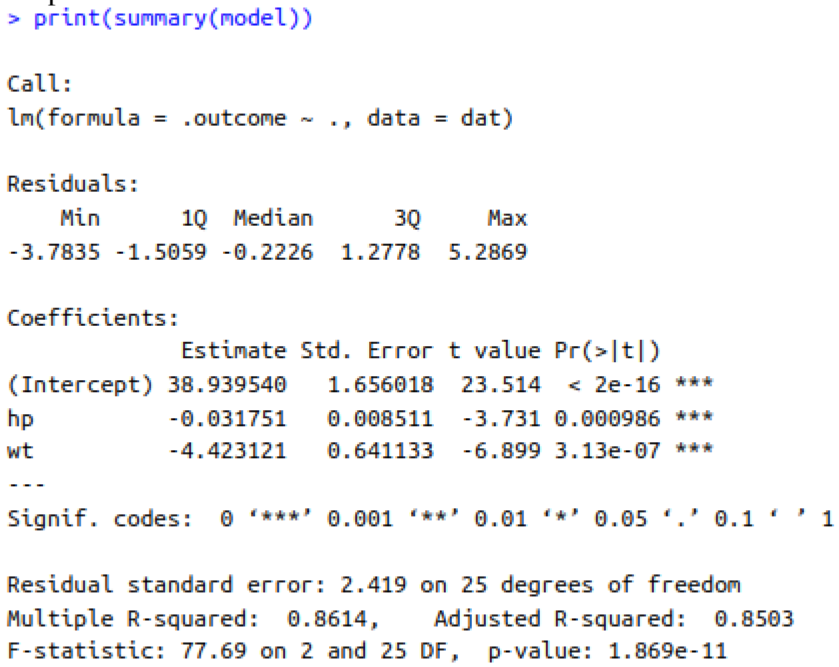
```
> print(result)
   demand mean_demand
    <num>       <num>
1:    8.3         8.3
2:   10.3        10.3
3:   19.0        19.0
4:   16.0        16.0
5:   15.6        15.6
6:   19.8        19.8
```

5. caret

Code:

```
library(caret)
# Splitting data into training and testing sets
data <- mtcars
set.seed(163)
trainIndex <- createDataPartition(data$mpg, p = 0.8, list = FALSE)
train_data <- data[trainIndex, ]
test_data <- data[-trainIndex, ]
# Train a linear regression model
model <- train(mpg ~ hp + wt, data = train_data, method = "lm")
print(summary(model))
```

Output:



```
> print(summary(model))

Call:
lm(formula = .outcome ~ ., data = dat)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.7835 -1.5059 -0.2226  1.2778  5.2869 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 38.939540   1.656018  23.514  < 2e-16 ***
hp          -0.031751   0.008511  -3.731 0.000986 ***
wt          -4.423121   0.641133  -6.899 3.13e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.419 on 25 degrees of freedom
Multiple R-squared:  0.8614,	Adjusted R-squared:  0.8503 
F-statistic: 77.69 on 2 and 25 DF,  p-value: 1.869e-11
```