ST1131

Michael Yang

1 dplyr

1.1 filter

Pass in any number of conditions:

```
starwars %>% filter(skin_color == "light", eye_color == "brown")
```

is roughly equivalent to

```
starwars[starwars$skin_color == "light" & starwars$eye_color == "brown", ]
```

1.2 arrange

Order rows by column, tiebreak by next column

```
starwars %>% arrange(height, mass) #sort by height, tiebreak by mass

starwars %>% arrange(desc(height)) #sort descending
```

1.3 slice

```
starwars %>% slice(5:10)

starwars %>% slice_head(n = 3) # or slice_tail for last few rows

starwars %>% slice_sample(n = 5) # random sample

starwars %>%
  filter(!is.na(height)) %>%
  slice_max(height, n = 3) # get highest 3 rows by height
```

1.4 select

```
# Select columns by name
starwars %>% select(hair_color, skin_color, eye_color)

# Select all columns between hair_color and eye_color (inclusive)
starwars %>% select(hair_color:eye_color)

# Select all columns except those from hair_color to eye_color (inclusive)
starwars %>% select(!(hair_color:eye_color))

# Select all columns ending with color
starwars %>% select(ends_with("color"))

# Select variables (drops all other variables not explicitly mentioned)
starwars %>% select(home_world = homeworld)

# Rename variables
starwars %>% rename(home_world = homeworld)
```

Helper functions are `starts_with, ends_with, matches(), contains()`

1.5 mutate

Add new columns

```
starwars %>% mutate(height_m = height / 100)

# Add and select
starwars %>%
```

```r
  mutate(height_m = height / 100) %>%
  select(height_m, height, everything())

starwars %>%
  mutate(
    height_m = height / 100,
    BMI = mass / (height_m^2)
  ) %>%
  select(BMI, everything())

# Only keep new variables
starwars %>%
  mutate(
    height_m = height / 100,
    BMI = mass / (height_m^2),
    .keep = "none"
  )
```

1.6 relocate

Reorder columns

```r
starwars %>% relocate(sex:homeworld, .before = height)
```

1.7 summarise

Collapse data frame to a single row

```r
starwars %>% summarise(height = mean(height, na.rm = TRUE))

food_consumption %>%
  # Filter for rice food category
  filter(food_category == "rice") %>%
  # Summarize the mean_co2 and median_co2
  summarize(mean_co2 = mean(co2_emission),
            median_co2 = median(co2_emission))
```

1.8 count

```r
# Usage
# count(x, ..., wt = NULL, sort = FALSE, name = NULL)

starwars %>% count(species, sort = TRUE)
starwars %>% count(sex, gender, sort = TRUE)
starwars %>% count(birth_decade = round(birth_year, -1))
```

2 ggplot2

```
# Usage
ggplot(data = NULL, mapping = aes(), environment = parent.frame())

aes(x = ..., y = ..., colour = ..., size = ...)
```

`ggplot()` is used to construct the initial plot object, and is almost always followed by a plus sign (+) to add components to the plot.

```
+ geom_histogram(bars = 5)
+
```

3 Measures of Spread

```r
# Variance
var(food_consumption$consumption)

# SD
sd(food_consumption$consumption)

# Quartile
quartile(food_consumption$consumption)

# Quintile
quantile(food_consumption$co2_emission, probs = seq(0, 1, 0.2))
quantile(food_consumption$co2_emission, probs = c(0, 0.2, 0.4, 0.6, 0.8, 1.0))

# Decile
quantile(food_consumption$co2_emission, probs = seq(0, 1, 0.1))

# IQR
q1 <- quantile(food_consumption$co2_emission, 0.25) # 25 percentile
q3 <- quantile(food_consumption$co2_emission, 0.75) # 75 percentile
iqr <- q3 - q1

# Cutoff for outliers
lower <- q1 - 1.5 * iqr
upper <- q3 + 1.5 * iqr

# Filter to find outliers
food_consumption %>%
  filter(co2_emission < lower | co2_emission > upper) # note the | or condition instead of ,
```

4 Probability

```r
# Fixed seed
set.seed(5)

# Random sample
sales_count %>%
  sample_n(1)

# Sampling without replacement
sales_count %>%
  sample_n(2)

# Sample with replacement
sales_count %>%
  # Sampling with replacement is independent, without replacement is dependent
  sample_n(2, replace = TRUE)

# Count number of deals and probability of each deal
amir_deals %>%
  count(product) %>%
  mutate(prob = n / sum(n))

# mean
mean(x, trim = 0, na.rm = FALSE) #na.rm means whether or not to remove NA
```

5 Distributions

Law of large sizes: sample mean approaches theoretical mean as sample size increases

5.1 Discrete

```
rolls_10 <- die %>%
  sample_n(10, replace = TRUE) %>%
  ggplots(aes(n)) +
    geom_histogram(bins = 6)

expected_val <- sum(size_distribution$group_size *
                    size_distribution$probability)


# Calculate probability of picking group of 4 or more
size_distribution %>%
  # Filter for groups of 4 or larger
  filter(group_size >= 4) %>%
  # Calculate prob_4_or_more by taking sum of probabilities
  summarize(prob_4_or_more = sum(probability))
```

5.2 Continuous

```
# P(<= 7)
punif(7, min = 0, max = 12)

# P(> 7)
punif(7, min = 0, max = 12, lower.tail = FALSE)

# P(4 <= p <= 7)
punif(7, min = 0, max = 12) - punif(4, min = 0, max = 12)

# Set random seed to 334
set.seed(334)

# Generate 1000 wait times between 0 and 30 mins, save in time column
wait_times %>%
  mutate(time = runif(1000, min = 0, max = 30)) %>%
  # Create a histogram of simulated times
  ggplot(aes(time)) +
  geom_histogram(bins = 30)

# Simulate n generations of values between 0 to 30
runif(n, min = 0, max = 30)
```

5.3 Binomial

```
# Simulating trials
rbinom(trials, coins, probability of success)
rbinom(8, 1, 0.5) # 8 flips of 1 coin with 50% success
rbinom(1, 8, 0.5) # 1 flip of 8 coins with 50% success
rbinom(10, 3, 0.5) # 10 flips of 3 coins

# Calculating probability
# dbinom(num heads, num trials, prob of heads)
dbinom(7, 10, 0.5) # P(heads = 7)
pbinom(7, 10, 0.5) # P(heads <= 7)
pbinom(7, 10, 0.5, lower.tail = FALSE) # P(heads > 7)
1 - pbinom(7, 10, 0.5) # P(heads > 7)
```

```r

# Simulate 52 weeks of 3 deals
deals <- rbinom(52, 3, 0.3)
# Calculate mean deals won per week
mean(deals)

# Probability of closing 3 out of 3 deals with 30% success
dbinom(3, 3, 0.3)

# Probability of closing <= 1 deal out of 3 deals
pbinom(1, 3, 0.3)

# Probability of closing > 1 deal out of 3 deals
pbinom(1, 3, 0.3, lower.tail = FALSE)

# Expected number won with 30% win rate
won_30pct <- 3 * 0.3
```

Expected value $= n \times p$
Each trial must be independent for binomial distribution

5.4 Normal

```r
# Percentage of values <= 154
pnorm(154, mean = 161, sd = 7)

# values > 154
pnorm(154, mean = 161, sd = 7, lower.tail = FALSE)

# 154 <= value <= 157
pnorm(157, mean = 161, sd = 7) - pnorm(154, mean = 161, sd = 7)

# value that 90% of values <=
qnorm(0.9, mean = 161, sd = 7)

# value that 90% of values >=
qnorm(0.9, mean = 161, sd = 7, lower.tail = FALSE)

# Generate 10 random numbers
rnorm(10, mean = 161, sd = 7)

# Compare performance over 2 quarters
pnorm(1000, mean = 5000, sd = 2000, lower.tail = FALSE)
pnorm(1000, mean = 5600, sd = 2600, lower.tail = FALSE)
```

5.5 Central Limit Theorem

```r
die <- c(1, 2, 3, 4, 5, 6)
sample_5 <- sample(die, 5, replace = TRUE)
sample(die, 5, replace = TRUE) %>% mean()

# Replicating 10 times of random sample of 5
sample_mean <- replicate(10, sample(die, 5, replace = TRUE) %>% mean())
sample_sd <- replicate(10, sample(die, 5, replace = TRUE) %>% sd())
```

Central Limit Theorem: sample distribution becomes closer to normal distribution as number of trials increase. (*samples should be random and independent)

5.6 Poisson

Events appear to happen at a certain rate, but completely at random.

E.g. number of animals adopted form animal shelters per week / Number of people arriving at a restaurant per hour / number of earthquakes per year
Poisson distribution = probability of some # of events occuring over a fixed period of time, e.g. probability of < 20 earthquakes in California per year

λ = average number of events per time interval e.g. average number of adoptions per week. (Also the expected value of the distribution)

Lambda is the distribution's peak.

```
# P(= 5), average = 8 (exact value)
dpois(5, lambda = 8)

# P(<= 5), average = 8 (less than or equal to)
ppois(5, lambda = 8)

# P(> 5), avg = 8
ppois(5, lambda = 8, lower.tail = FALSE)

# Sampling from Poisson distribution
rpois(10, lambda = 8)
```

CLT still applies to sample means of poisson distribution

5.7 Exponential

Represents probability of certain time passing between Poisson events e.g. probability of ¿ 1 day between adoption, probability of ¡ 10 mins between restaurants, probability of 6-8 months between earthquakes. Also uses lambda (Rate)

Continuous (time)

```
# P(wait < 1min)
pexp(1, rate = 0.5)

# P(wait > 4min)
pexp(4, rate = 0.5, lower.tail = FALSE)

# P(1min < wait < 4min)
pexp(4, rate = 0.5) - pexp(1, rate = 0.5)

# Average response time = 2.5 hours from receiving request
# P(< 1 hour to respond)
pexp(1, rate = 1 / 2.5)

# P(> 4hrs to respond)
pexp(1, rate = 0.4, lower.tail = FALSE)

# P(3 < wait < 4 hrs to respond)
pexp(4, rate = 0.4) - pexp(3, rate = 0.4)
```

Expected time = $1/\lambda$

5.8 T Distribution

Same shape as normal distribution, but different shaped tails (observations more/less likely to fall further than/-closer to mean)

Degrees of freedom (df) affect thickness of tails. Lower df = thicker tails, higher sd. Higher df = closer to normal distribution

T distribution is not skewed.

5.9 Log-normal Distribution

Skewed distributions e.g. length of chess games, adult blood pressures, number of hospitalizations during 2003 SARS outbreak

6 Correlation

6.1 Relationship

positive correlation = same direction, negative correlation = opposite direction

```r
# Visualizing relationships
ggplot(df, aes(x, y)) +
  geom_point +
  geom_smooth(method = "lm", se = FALSE) # lm for linear trend line, se = false for no error margin

# Computing correlation
# Usage: corr(vec1, vec2)
cor(df$x, df$y)

cor(df$x, df$y, use = "pairwise.complete.obs") # ignore NA values, otherwise corr returns NA
```

If correlation = 0, there is no relationship and the points will appear to be randomly scattered.

Correlation can be written as r.

If the correlation has a high magnitude, the data points will be clustered closely around a line.

Plot of A against B means A is on the y-axis while B is on the x-axis.

6.2 Caveats of Correlation

Can't account for non-linear relationships e.g. quadratic (U shaped). Correlation shows linear relationships only. Always **visualize** the data.

Log transformation can be used to normalize highly skewed distributions.

```r
msleep %>%
  mutate(log_bodywt = log(bodywt)) %>%
  ggplot(...) + ...
```

Other transformations: square root `sqrt(x)`, reciprocal `1 / x`

Correlation does not imply causation. x can be correlated with y but x does not *cause* y.
Spurious correlation: two variables appear to be related but only due to a third, unseen factor, i.e. **confounder** / **lurking variable**.
E.g. holidays are associated with higher retail sales, but it is actually because of more holiday deals (**confounder**)

6.3 Experimental Design

Experiments aim to answer: what is the effect of **treatment** on the **response**? Treatment = explanatory/independent variable Response = dependent variable

E.g. what is the effect of an advertisement on the number of products purchased?
Treatment: advertisement Response: number of products purchased

Controlled experiments - participants are assigned at random to either treatment group (receives treatment) or control group (no treatment).

Groups should be **comparable** so that causation can be inferred. E.g. if average age of control group = 25 while average age of treatment group is 50, age could be a confounder.

Gold standard of experiments:

- ◇ Randomized controlled trial (assign to groups randomly)
- ◇ Placebo - resembles treatment, but has no effect, so participants don't know which group they are in, so causation is due to treatment, not the idea of the treatment (common in clinical trials for drug testing e.g. sugar pills)
- ◇ Double blind trial - person adminstering treatment also doesn't know if they are administering the actual treatment or placebo

Fewer opportunies for bias = more reliable response

Observational studies - participants assign themselves. Answers questions that are not conducive to a controlled experiment (e.g. can't force people to smoke or have a disease)

However, observational studies only establish association, not causation.

Longitudinal vs Cross-sectional study
Longitudinal: participants followed over a period of time to examine effect of treatment on response
Cross-sectional: data on participants is collected from a single snapshot in time

Longitudinal - more expensive, results take longer. Cross-sectional - cheaper, more convenient