interview_test.Rmd

Aileen

2025-11-14

Part 1

Task

Determine the number of sock pairs from the following pile:

pile = [3,3,1,2,3,2,1,3,2,3,1,3,3,1,3,1,1,2,2,2]

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(tibble)
library(purrr)
library(knitr)
```

```r
# Set pile
pile <- c(3,3,1,2,3,2,1,3,2,3,1,3,3,1,3,1,1,2,2,2)

colour_freq <- table(pile) #table() gives a summary of sock frequencies

colour_freq
```

```
## pile
## 1 2 3
## 6 6 8
```

```r
n_pairs <- colour_freq %/% 2 %>%   # divide the frequencies by 2 to get complete pairs
  sum()   # get sum of sock pairs

print(paste0("Total pairs of socks = ", n_pairs))
```

```
## [1] "Total pairs of socks = 10"
```

Create a function for next part:

```
# Create function get_n_pairs()
get_n_pairs <- function(pile){

  colour_freq <- table(pile)

  n_pairs <- colour_freq %/% 2 %>%
    sum()

  return(n_pairs)

}

# Apply function
get_n_pairs(pile)
```

```
## [1] 10
```

Part 2 - Simulation

Create a simulated dataset with 100 independent samplings, from n = 3 different colours of socks, with a sample size of 20, and determine the number of pairs in each pile.

```
# Set seed
set.seed(123)

# Create simulated dataset for n=3

# generate random values from 1-3 sock colours, 20 times per pile, and repeated 100 times to generate 1
sock_colours <- replicate(100, round(runif(20, 1,3),0), simplify = FALSE)

# label each pile
pile_num <- 1:100

# combine into a tibble dataframe

sim_dataset_n3 <- tibble(
  pile_colours = sock_colours,
  pile_num = pile_num)

# print simulated dataset
sim_dataset_n3 %>%
  head() %>%
  knitr::kable()
```

pile_colours	pile_num
2, 3, 2, 3, 3, 1, 2, 3, 2, 2, 3, 2, 2, 2, 1, 3, 1, 1, 2, 3	1

pile_colours	pile_num
3, 2, 2, 3, 2, 2, 2, 2, 2, 1, 3, 3, 2, 3, 1, 2, 3, 1, 2, 1	2
1, 2, 2, 2, 1, 1, 1, 2, 2, 3, 1, 2, 3, 1, 2, 1, 1, 3, 3, 2	3
2, 1, 2, 2, 3, 2, 3, 3, 3, 2, 3, 2, 2, 1, 2, 1, 2, 2, 2, 1	4
1, 2, 2, 3, 1, 2, 3, 3, 3, 1, 1, 2, 2, 2, 2, 1, 3, 1, 2, 2	5
2, 2, 2, 3, 2, 3, 3, 2, 2, 1, 3, 2, 1, 3, 2, 1, 2, 3, 2, 2	6

```r
# Apply get_n_pairs() function to each list in pile_colours column to get total pairs (n_pairs column)

sim_dataset_n3_totalsocks <- sim_dataset_n3 %>%
  dplyr::mutate(n_pairs = purrr::map_int(pile_colours, get_n_pairs))


sim_dataset_n3_totalsocks %>%
  head() %>%
  knitr::kable()
```

pile_colours	pile_num	n_pairs
2, 3, 2, 3, 3, 1, 2, 3, 2, 2, 3, 2, 2, 2, 1, 3, 1, 1, 2, 3	1	9
3, 2, 2, 3, 2, 2, 2, 2, 2, 1, 3, 3, 2, 3, 1, 2, 3, 1, 2, 1	2	10
1, 2, 2, 2, 1, 1, 1, 2, 2, 3, 1, 2, 3, 1, 2, 1, 1, 3, 3, 2	3	10
2, 1, 2, 2, 3, 2, 3, 3, 3, 2, 3, 2, 2, 1, 2, 1, 2, 2, 2, 1	4	9
1, 2, 2, 3, 1, 2, 3, 3, 3, 1, 1, 2, 2, 2, 2, 1, 3, 1, 2, 2	5	9
2, 2, 2, 3, 2, 3, 3, 2, 2, 1, 3, 2, 1, 3, 2, 1, 2, 3, 2, 2	6	9

Repeat the sampling for different number of sock colours in each pool from n = 3 to n = 18 with step size of 3.

```r
# Set seed
set.seed(123)

# Generate new dataset with more sock colours per pile
n_colours <- seq(3,18,3) # get number of sock colours from 3 to 18 with step size of 3

# make empty list for dataset
sock_colours <- list()

# populate with sock colours (20 colours per pile)
for (n in n_colours){
  sock_colours <- c(sock_colours, replicate(100, round(runif(20, 1, n),0), simplify = FALSE))
}

# annotate piles
n_colour <- rep(seq(3,18,3), each = 100)
pile_num <- rep(1:100, times = 6)

# combine into tibble dataset

sim_dataset <- tibble(
  sock_colours,
  n_colour,
  pile_num
)

sim_dataset %>%
  head(20) %>%
  knitr::kable()
```

sock_colours	n_colour	pile_num
2, 3, 2, 3, 3, 1, 2, 3, 2, 2, 3, 2, 2, 2, 1, 3, 1, 1, 2, 3	3	1
3, 2, 2, 3, 2, 2, 2, 2, 2, 1, 3, 3, 2, 3, 1, 2, 3, 1, 2, 1	3	2
1, 2, 2, 2, 1, 1, 1, 2, 2, 3, 1, 2, 3, 1, 2, 1, 1, 3, 3, 2	3	3
2, 1, 2, 2, 3, 2, 3, 3, 3, 2, 3, 2, 2, 1, 2, 1, 2, 2, 2, 1	3	4
1, 2, 2, 3, 1, 2, 3, 3, 3, 1, 1, 2, 2, 2, 2, 1, 3, 1, 2, 2	3	5
2, 2, 2, 3, 2, 3, 3, 2, 2, 1, 3, 2, 1, 3, 2, 1, 2, 3, 2, 2	3	6
2, 2, 2, 1, 2, 3, 1, 1, 1, 2, 2, 3, 2, 2, 2, 2, 3, 3, 3, 2	3	7
2, 2, 1, 1, 3, 1, 1, 1, 1, 2, 3, 2, 2, 1, 1, 2, 2, 1, 2, 1	3	8
2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 1, 3, 2, 2, 2, 2, 2, 3, 2	3	9
3, 2, 2, 2, 2, 2, 2, 2, 3, 3, 2, 2, 3, 2, 3, 2, 2, 2, 1, 2	3	10
1, 3, 2, 2, 2, 3, 2, 2, 1, 1, 2, 2, 1, 2, 1, 2, 2, 2, 3, 3	3	11
2, 3, 2, 2, 1, 2, 2, 2, 2, 3, 2, 2, 2, 1, 2, 2, 1, 3, 1, 3	3	12
2, 2, 1, 2, 2, 2, 2, 3, 3, 2, 2, 1, 2, 2, 2, 3, 1, 2, 2, 3	3	13
3, 3, 2, 3, 2, 2, 2, 2, 1, 2, 3, 1, 1, 1, 3, 2, 3, 2, 1, 2	3	14
3, 1, 2, 1, 1, 2, 1, 1, 1, 2, 2, 1, 1, 3, 3, 3, 3, 1, 1, 3	3	15
3, 1, 3, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 1, 2, 3, 3, 1, 2, 3	3	16
3, 2, 1, 2, 1, 1, 3, 1, 2, 3, 2, 2, 2, 3, 2, 2, 2, 2, 2, 3	3	17
2, 1, 2, 1, 2, 2, 3, 2, 1, 2, 1, 3, 2, 1, 2, 3, 2, 2, 2, 3	3	18
1, 1, 3, 2, 2, 3, 1, 1, 2, 2, 2, 2, 3, 2, 2, 3, 3, 1, 2, 3	3	19
2, 3, 2, 3, 2, 3, 1, 2, 1, 1, 3, 2, 2, 3, 1, 1, 2, 1, 1, 3	3	20

```r
# calculate total pairs
sim_dataset_results <- sim_dataset %>%
  dplyr::mutate(n_pairs = purrr::map_int(sock_colours, get_n_pairs))

# show table summary (without sock_colours column)
sim_dataset_results %>%
  select(-c(sock_colours)) %>%
  summary() %>%
  knitr::kable("html")
```

n_colour

pile_num

n_pairs

Min. : 3.0

Min. : 1.00

Min. : 4.000

1st Qu.: 6.0

1st Qu.: 25.75

1st Qu.: 7.000

Median :10.5

Median : 50.50

Median : 8.000

Mean :10.5

Mean : 50.50

Mean : 7.528

3rd Qu.:15.0

3rd Qu.: 75.25

3rd Qu.: 9.000

Max. :18.0

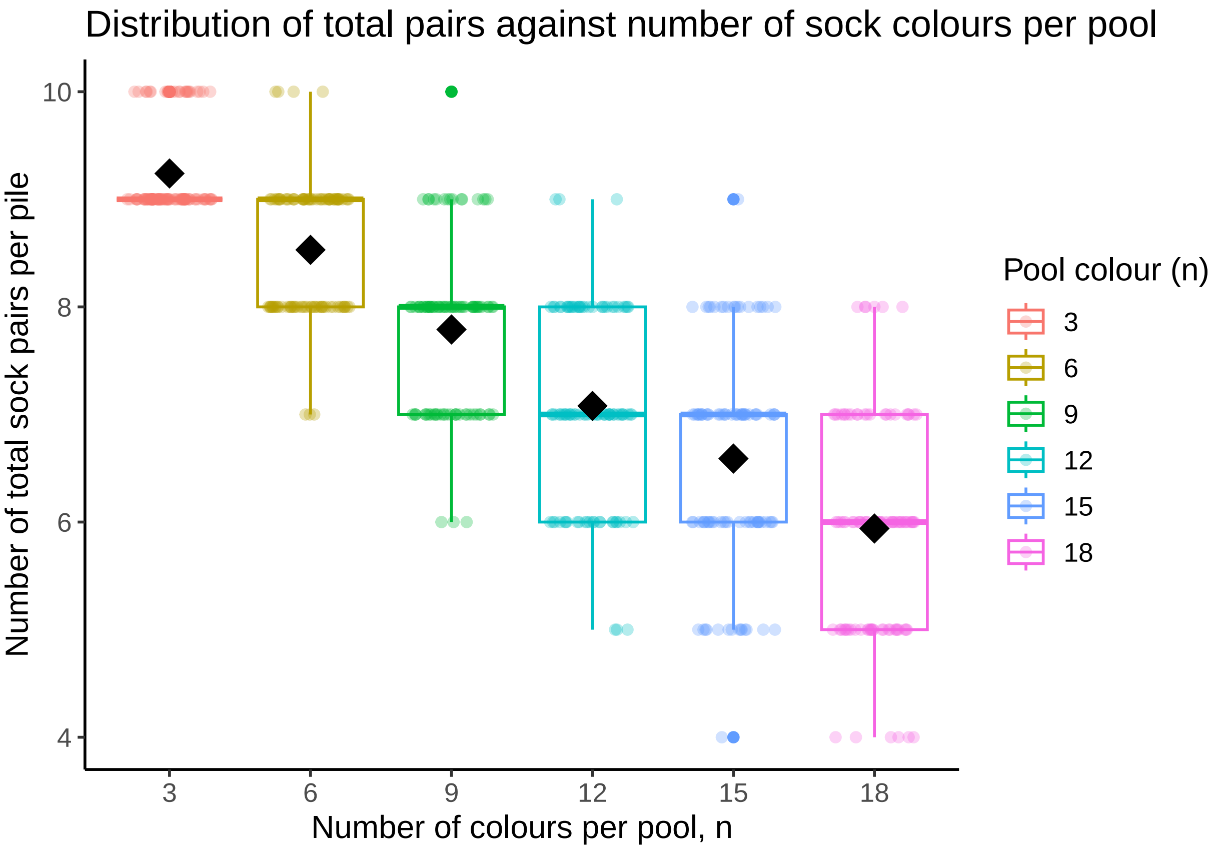
Max. :100.00

Max. :10.000

Visualise how the number of pairs varies as the pools of different colours increase.

```r
# Create boxplot to illustrate distribution of total pairs per n_colour

sim_dataset_results %>%
  dplyr::mutate(n_colour = as.factor(n_colour)) %>%
  ggplot(aes(x = n_colour, y = n_pairs, colour = n_colour)) +
  geom_boxplot() +
  geom_jitter(height = 0, width = 0.3, alpha = 0.3)+
  stat_summary(fun.y=mean, geom="point", shape=18, size=5, color="black")+
  theme_classic()+
  theme(text = element_text(size = 12))+
  labs(title = "Distribution of total pairs against number of sock colours per pool",
    x = "Number of colours per pool, n",
       y = "Number of total sock pairs per pile",
       colour = "Pool colour (n)")
```

```
## Warning: The ‘fun.y‘ argument of ‘stat_summary()‘ is deprecated as of ggplot2 3.3.0.
## i Please use the ‘fun‘ argument instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



The boxplot above shows the distribution of total pairs in each sampled pile, with the average total pairs per n_colour (number of colours in pool) shown with the black diamond point.

1. Is there a relationship between the number of colours to select from and the average number of pairs.

There is a visual negative relationship between the number of colours in the pool and the average number of pairs in the sampled pile.

2. Perform a statistical test to examine this.

There are two statistical approaches I can take, one where I treat the number of colours in the pool as an independent "categorical" factor, and test whether the mean number of pairs are different between the groups (ANOVA), or I use linear regression to estimate how the number of pairs changes as the number of colours increases.

Here, I'm choosing to use linear regression to estimate the trend (relationship between these two variables), and how strongly the number of colours predicts the number of pairs in a pile, rather than just test the difference in mean pairs between the groups.

The null hypothesis is that there is *no relationship* between the number of colours in the pull and the number of pairs in the pile, while the alternate hypothesis is that the number of colours *influences* the number of pairs.

```r
# fit linear regression model
lm_sim <- lm(n_pairs ~ n_colour, data = sim_dataset_results)

# get coefficients
summary(lm_sim)
```

```
##
```

```
## Call:
## lm(formula = n_pairs ~ n_colour, data = sim_dataset_results)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5413 -0.5153 -0.1733  0.4847  2.4587
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.831333   0.075549  130.13   <2e-16 ***
## n_colour    -0.219333   0.006466  -33.92   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.8115 on 598 degrees of freedom
## Multiple R-squared:  0.658,  Adjusted R-squared:  0.6574
## F-statistic:  1150 on 1 and 598 DF,  p-value: < 2.2e-16
```

The estimated coefficient for n_colour shows a *negative relationship* (Estimate = -0.219) between total pairs (n_pairs) and number of colours (n_colour). Specifically, this model estimates that for each additional colour in the pool, there is an expected 0.219 decrease in the number of pairs of socks in the pile.

The p-value is also very small (<2e-16), so I reject the null hypothesis that there is no relationship between number of colours and number of pairs in the pile.

The model fit (adjusted R-squared value) is 0.66, which in this case suggests that the number of colours in the pool is a strong predictor for number of pairs in this model (explains 66% of variance in total pairs of socks).

Thus, I would say that there is a strong negative relationship between the number of colours in the pool and the number of sock pairs in a sampled pile.

3. Given these results train a statistical model and estimate how many pairs there will be when n = 30 (any type of model or machine learning algorithm). Is this answer accurate?

From the above question, we already have a linear regression model trained from the simulated dataset from n = 3 to n = 18. Thus, we can use the predict() function to extrapolate how many pairs there will be when n = 30.

```
newpool <- data.frame(n_colour = 30)
predict(lm_sim, newdata = newpool, interval = "predict")
```

```
##        fit      lwr      upr
## 1 3.251333 1.637102 4.865565
```

In this model (lm_sim), the 95% prediction interval of number of pairs when n = 30 is between 4.89 and 1.75 pairs.

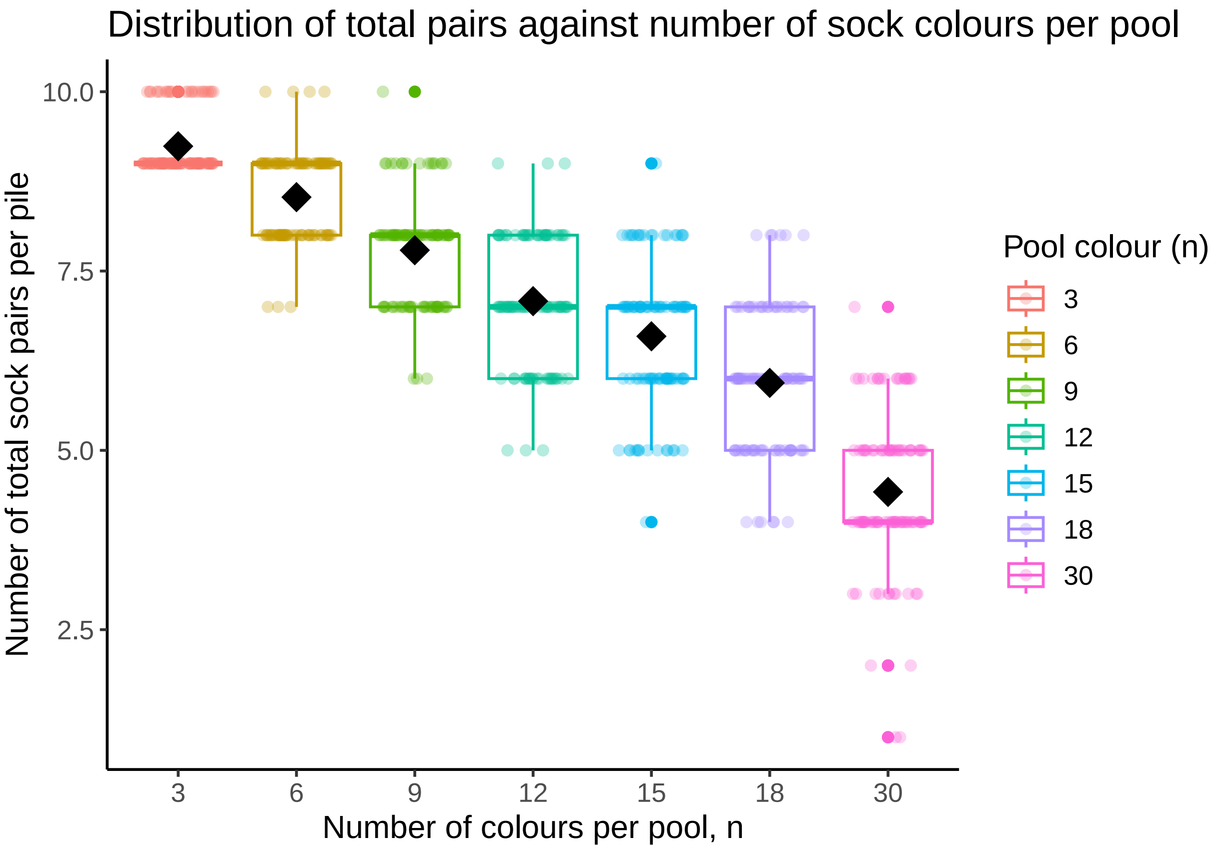
To test the accuracy of this model, I will generate a simulated test dataset where n = 30 and quantify the differences between the predicted mean and observed mean of total pairs.

```
# Set seed
set.seed(123)

# generate test dataset
sock_colours <- replicate(100, round(runif(20, 1,30),0), simplify = FALSE)
```

```r
pile_num <- 1:100
n_colour <- rep(30, times=100)

sim_dataset_n30 <- tibble(
  sock_colours,
  pile_num,
  n_colour
)

test_data <- sim_dataset_n30 %>%
  dplyr::mutate(total_pairs = purrr::map_int(sock_colours, get_n_pairs))

test_data %>%
  head() %>%
  knitr::kable()
```

sock_colours	pile_num	n_colour	total_pairs
9, 24, 13, 27, 28, 2, 16, 27, 17, 14, 29, 14, 21, 18, 4, 27, 8, 2, 11, 29	1	30	4
27, 21, 20, 30, 20, 22, 17, 18, 9, 5, 29, 27, 21, 24, 2, 15, 23, 7, 10, 8	2	30	3
5, 13, 13, 12, 5, 5, 8, 15, 9, 26, 2, 14, 24, 5, 17, 7, 5, 23, 27, 12	3	30	4
20, 4, 12, 9, 25, 14, 24, 25, 24, 14, 23, 19, 22, 1, 15, 7, 12, 19, 11, 4	4	30	6
8, 20, 13, 24, 4, 14, 30, 27, 27, 6, 5, 20, 11, 20, 10, 6, 24, 4, 15, 16	5	30	5
18, 11, 15, 29, 15, 27, 28, 19, 13, 5, 28, 10, 3, 28, 22, 5, 17, 29, 18, 13	6	30	6

```r
# add to earlier box plot
sim_dataset_results %>%
  bind_rows(test_data %>% dplyr::rename(n_pairs = total_pairs)) %>%
  dplyr::mutate(n_colour = as.factor(n_colour)) %>%
  ggplot(aes(x = n_colour, y = n_pairs, colour = n_colour)) +
  geom_boxplot() +
  geom_jitter(height = 0, width = 0.3, alpha = 0.3)+
  stat_summary(fun.y=mean, geom="point", shape=18, size=5, color="black")+
  theme_classic()+
  theme(text = element_text(size = 12))+
  labs(title = "Distribution of total pairs against number of sock colours per pool",
    x = "Number of colours per pool, n",
       y = "Number of total sock pairs per pile",
       colour = "Pool colour (n)")
```



```r
# Add predicted total pairs to simulated dataset
predict_data <- test_data %>%
  dplyr::mutate(total_pairs_pred = predict(lm_sim, newdata = .))

# Compare means between observed and predicted total pairs
predict_data %>%
  summarise(
    observed_mean = mean(total_pairs),
    predicted_mean = mean(total_pairs_pred),
    RMSE = sqrt(mean((total_pairs - total_pairs_pred)^2))
  )
```

```
## # A tibble: 1 x 3
##   observed_mean predicted_mean  RMSE
##           <dbl>          <dbl> <dbl>
## 1          4.42           3.25  1.63
```

This model predicts that at n = 30 colours, the average number of pairs in a pile of 20 socks would be 3.32 pairs. Comparing this with a newly generated test dataset for n = 30, we observe an average of 4.42 pairs. This suggests that the model underestimates the number of pairs when n values are high. This is further confirmed by the RMSE (root mean squared error) at 1.69 pairs, which is a significant prediction error considering that this is around 38% of the observed mean.

In retrospect, a linear regression model assumes a constant rate of change, while the sock problem is actually non-linearly decreasing. As the number of colours increases, the probability of pulling matching pairs decreases rapidly then levels off, producing an asymptotic curve. Thus, a linear regression model trained between n = 3 - 18 would underpredict at larger n values like n = 30.

A non-linear regression model may be more appropriate to describe this process.