Day 26 - Single mean (non-parametric)

Learning outcomes

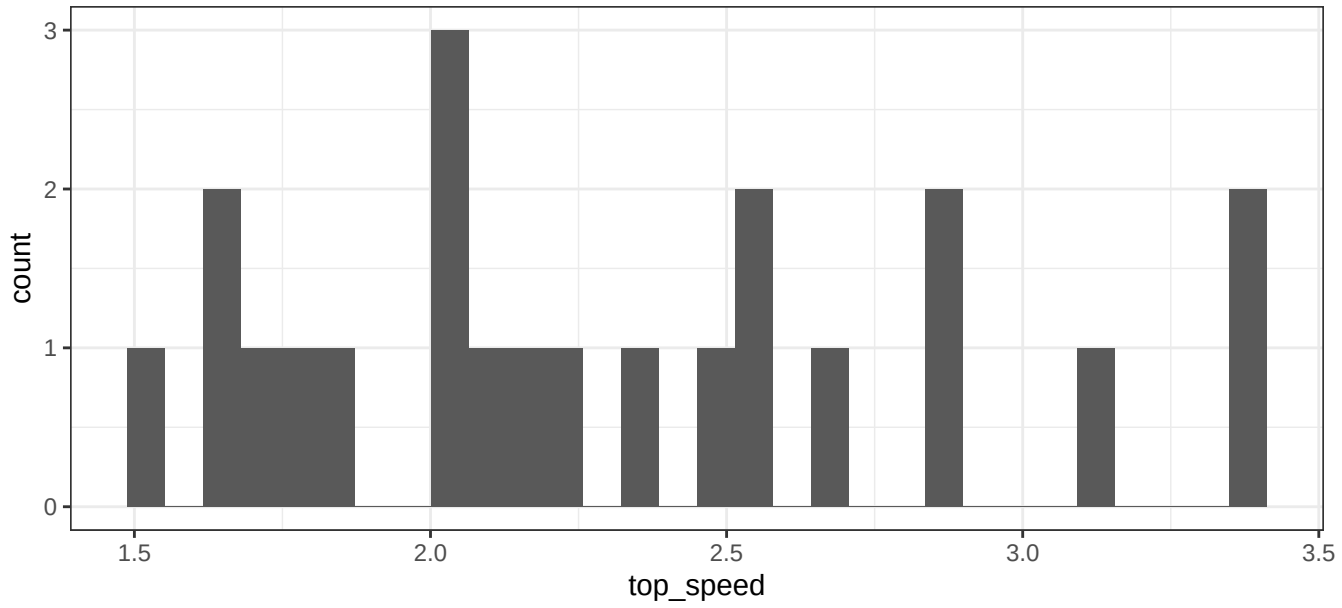
In this section, we explore how to formally make inference about a single quantitative variable using simulation-based methods.

Motivating example

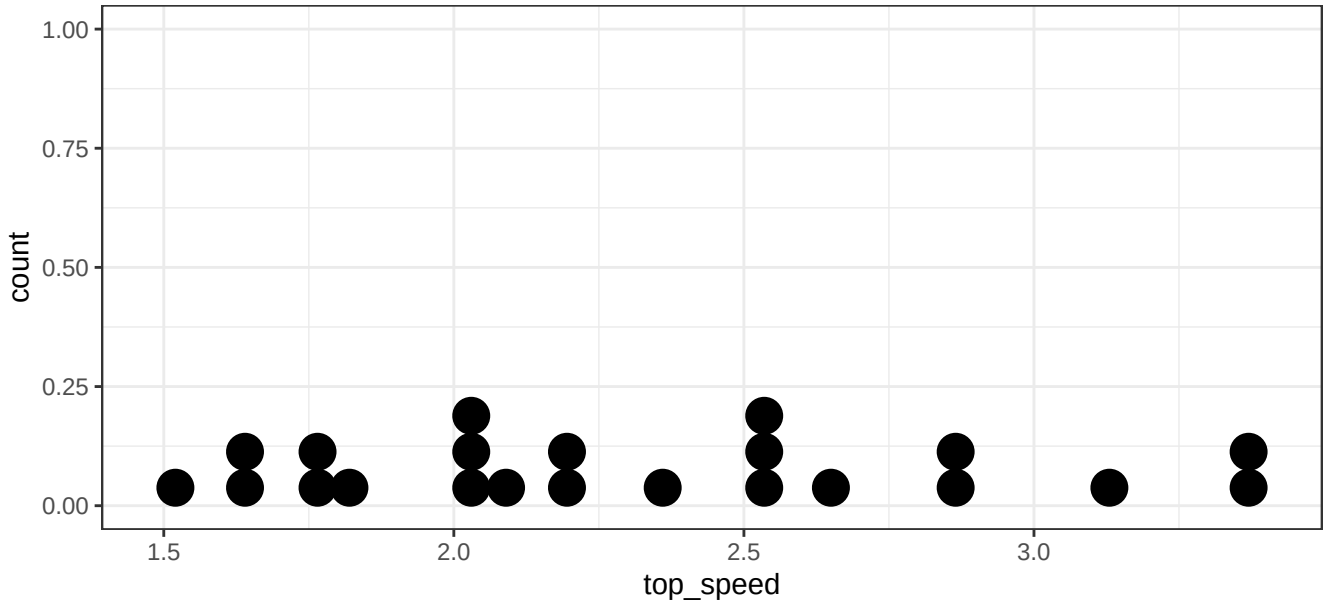
Lizard race Are western fence lizards faster than sagebrush lizards? Sagebrush lizards, a species of lizard commonly found at mid to high elevations in the Western United States, have been measured to have an average speed of 1.9 meters per second. Researchers are interested in determining whether western fence lizards, a species that occupies a similar niche, are faster than sagebrush lizards, on average. To test their hypothesis, speed measurements (in m/s) were collected on 22 western fence lizards. Is there evidence that western fence lizards are faster than sagebrush lizards, assuming that sagebrush lizards have an average speed of 1.9 m/s?

```
# load data
library(openintro)
data("lizard_run")
lizard <- lizard_run %>% filter(common_name == "Western fence lizard")

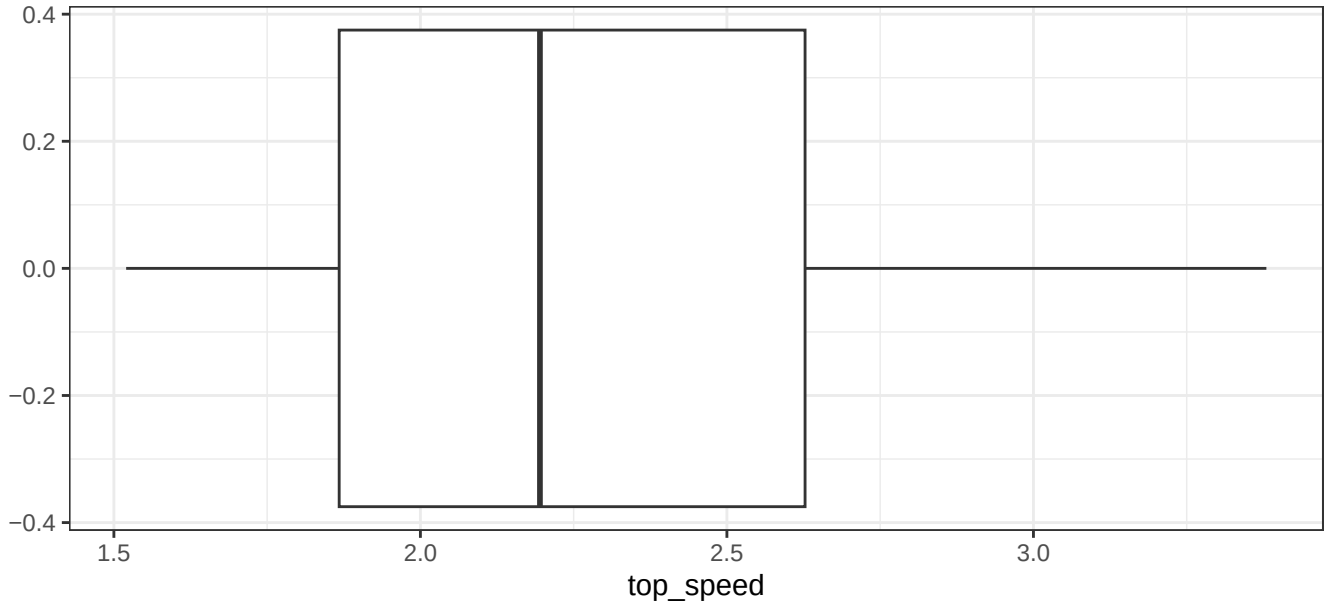
# some plots
## histogram
lizard %>%
  ggplot() +
  geom_histogram(aes(x = top_speed)) +
  theme_bw()
```




```
## dot plot
lizard %>%
  ggplot() +
  geom_dotplot(aes(x = top_speed)) +
  theme_bw()
```




```
## boxplot
lizard %>%
  ggplot() +
  geom_boxplot(aes(x = top_speed)) +
  theme_bw()
```



Review

What are the observational units for this study?

What variable is collected? What type of variable is it?

What is the research question?

Can we generalize the results of our inference to all western fence lizards?

Non-parametric inference for a single mean

- **Parameter of interest**
- **Sample statistic**
- **Null hypothesis**
- **Alternative hypothesis**
- **Assumptions**

Simulation *not* under the null hypothesis (confidence intervals)

> **Simulation not under the null**
>
> To estimate the variability in the sample mean, we randomly sample with _____ from the original data. This process is called _____.

Example: We could create the bootstrap distribution for the lizard example by the following process:

1. Label _____ cards with the _____ lizard speeds.
2. Resample, with _____, from the cards _____ times.
3. Calculate the bootstrap _____.
4. Repeat this process 10,000 times to generate the distribution of sample means.

```
# step 1: label the cards with the 22 lizard speeds
cards <- lizard$top_speed

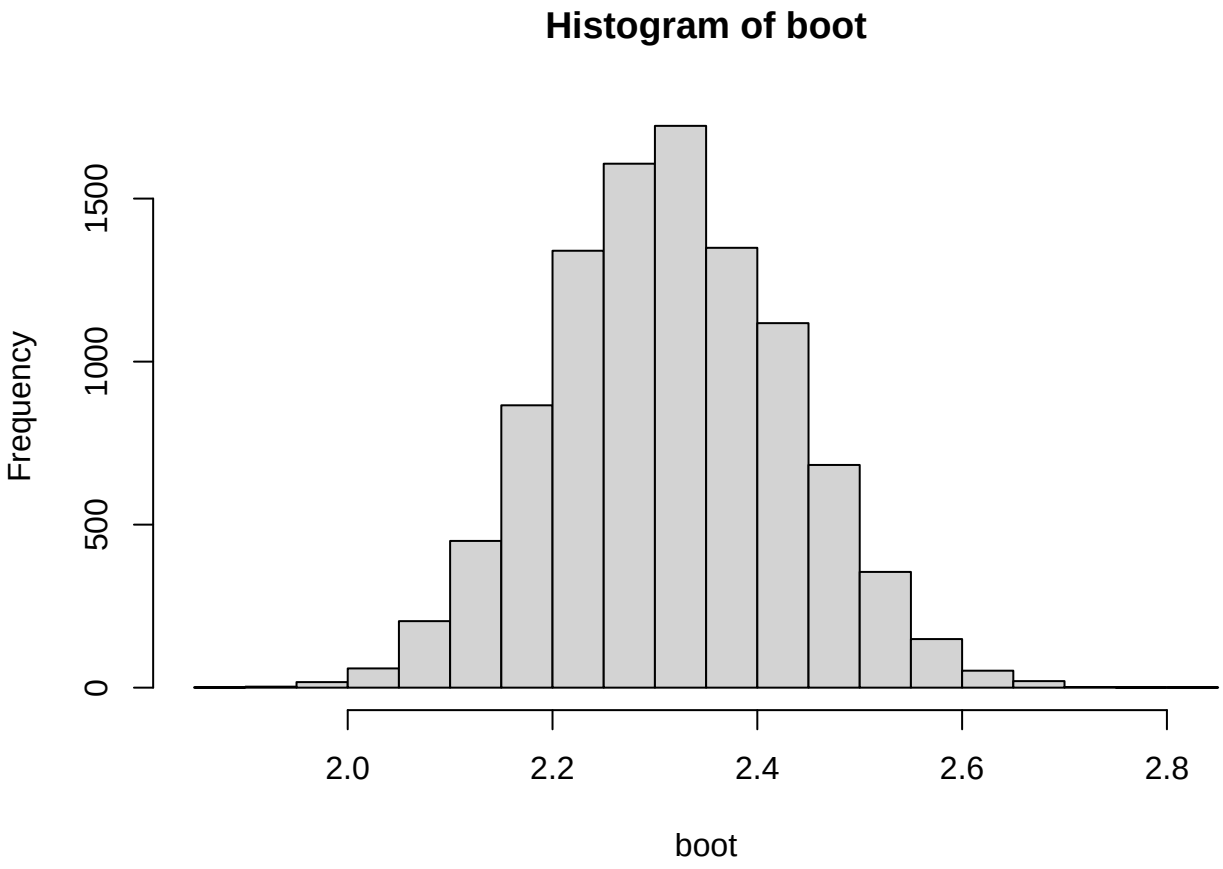
# step 2: resample with replacement 22 times
## first create storage
nsims <- 10000
boot <- matrix(NA, nsims, 1)
for(i in 1:nsims){
  resampled_cards <- sample(cards, size = 22, replace = T)
  bootstrap_mean <- mean(resampled_cards)
  boot[i,] <- bootstrap_mean
}
mean(boot)
```

```
[1] 2.315649
```

```
sd(boot) # standard error
```

```
[1] 0.1156301
```

```
hist(boot, breaks = 24)
```




```
quantile(boot, c(0.025, 0.975)) # 95% confidence interval
```

```
    2.5%    97.5%
2.093625 2.545466
```

Simulation under the null hypothesis (hypothesis testing)

> **Simulation under the null**
>
> Under the null hypothesis, we expect the mean of the _____ of the sample means to be equal to _____. Therefore, to simulate sample _____ under the _____, we shift the bootstrap distribution to have mean equal to _____.

Example: We could create the null distribution for the lizard example by the following process:

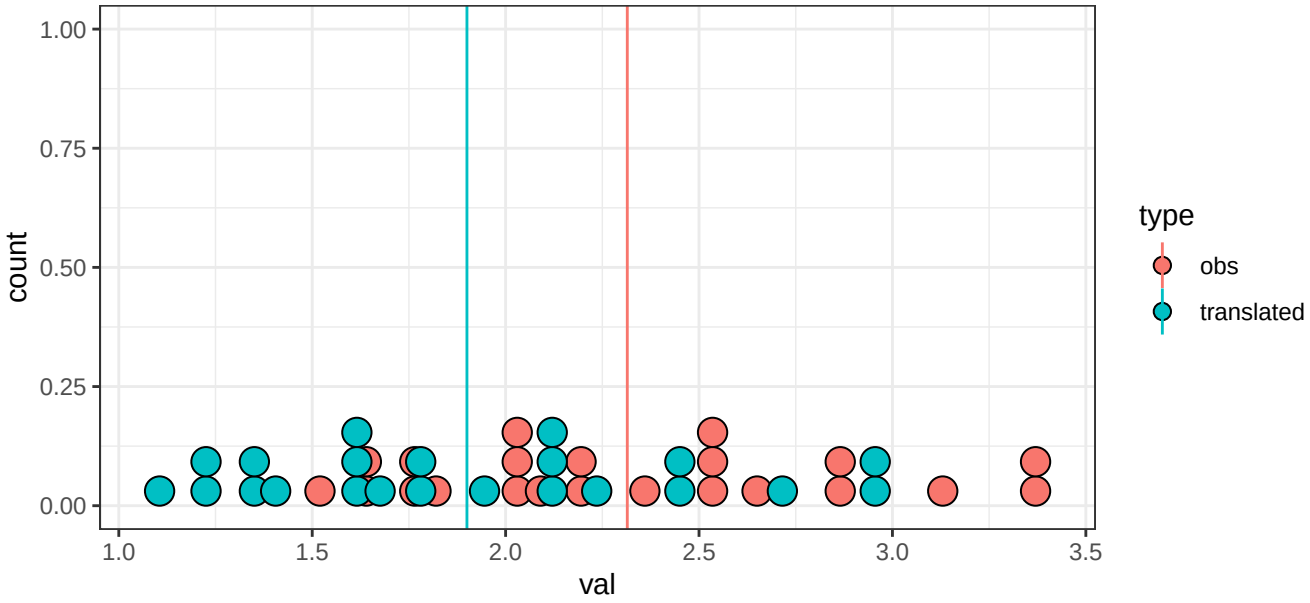
1. Label _____ cards with the _____ lizard speeds.
2. Resample, with _____, from the cards _____ times.
3. Calculate the bootstrap _____.
4. Repeat this process 10,000 times to generate the distribution of sample means.
5. Shift the distribution to have mean _____.

```
# demonstrate the process of moving the sample to have mean equal
tbl <- tibble(
  obs = lizard$top_speed
) %>%
  mutate(translated = obs + 1.9 - mean(obs)) %>%
  pivot_longer(obs:translated, names_to = "type", values_to = "val")

tbl %>% group_by(type) %>% summarize(mean = mean(val))
```

```
# A tibble: 2 x 2
  type        mean
  <chr>      <dbl>
1 obs         2.31
2 translated  1.9
```

```
tbl %>%
  ggplot() +
  geom_dotplot(aes(x = val, fill = type)) +
  geom_vline(
    data = tibble(val = c(mean(lizard$top_speed), 1.9), type = c("obs", "translated")),
    aes(xintercept = val, col = type)
  ) +
  theme_bw()
```


1.00
0.75
0.50
0.25
0.00
count
1.0
1.5
2.0
2.5
3.0
3.5
val
type
obs
translated