Week 1

AUTHOR
Daniel Poisel

PUBLISHED
October 21, 2024

```
library(tidyverse)
```

```
Warning: Paket 'stringr' wurde unter R Version 4.4.1 erstellt

── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✓ dplyr     1.1.4     ✓ readr     2.1.5
✓ forcats   1.0.0     ✓ stringr   1.5.1
✓ ggplot2   3.5.1     ✓ tibble    3.2.1
✓ lubridate 1.9.3     ✓ tidyr     1.3.1
✓ purrr     1.0.2
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
errors
```

```
# Custom plot theme
dark_theme_custom <- theme(
  panel.background = element_rect(fill = "#3a3a4f"),  # Brighter dark background colo
  plot.background = element_rect(fill = "#3a3a4f"),   # Same as panel background
  panel.grid.major = element_line(color = "#4b4b6b"), # Grid color slightly brighter
  panel.grid.minor = element_line(color = "#4b4b6b"),
  axis.text = element_text(color = "white", size = 12), # White axis text with larger
  axis.title = element_text(color = "white", size = 14), # White axis titles with lar
  legend.background = element_rect(fill = "#3a3a4f"), # Dark legend background
  legend.text = element_text(color = "white"),        # White legend text
  legend.title = element_text(color = "white"),       # White legend title
  plot.title = element_text(color = "white", hjust = 0.5), # Centered white plot titl
  panel.border = element_blank(),                     # Remove panel border
  axis.line = element_line(color = "white")           # White axis lines
)
```

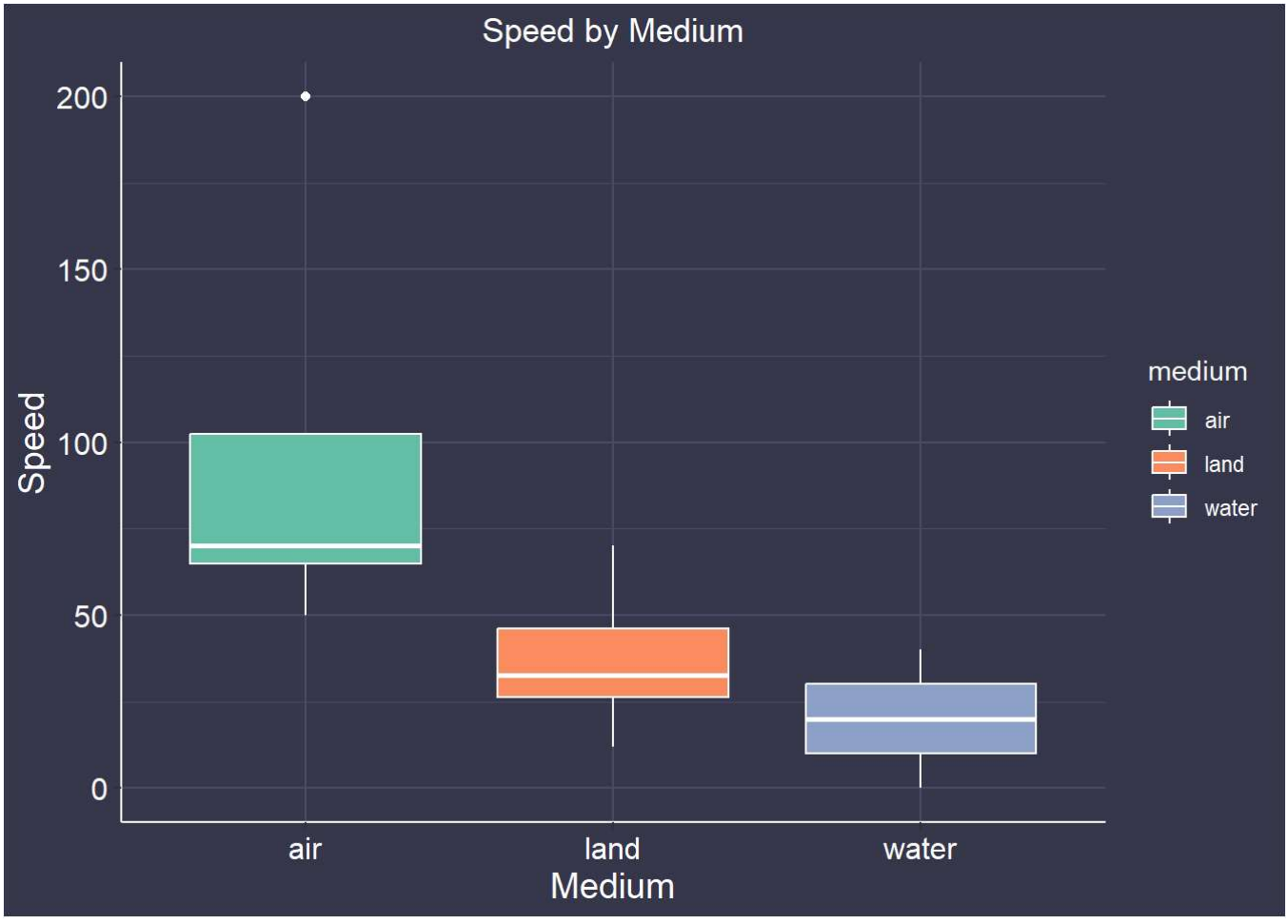
Task 1

```
# Create the dataframe
data <- tibble(
  species = c("swift", "falcon", "goose", "starling", "cheetah", "horse", "hippo", "m
  speed = c(200.00, 70.00, 70.00, 50.00, 70.00, 50.00, 30.00, 25.00, 12.00, 35.00, 40
  medium = factor(c("air", "air", "air", "air", "land", "land", "land", "land", "land
  weight = c(0.02, 0.70, 2.20, 0.05, 50.00, 450.00, 2500.00, 80.00, 0.60, 4.00, 600.0
)

# View the dataframe
print(data)
```

```
# A tibble: 12 × 4
   species   speed medium  weight
   <chr>     <dbl> <fct>    <dbl>
 1 swift    200    air       0.02
 2 falcon    70    air       0.7
 3 goose     70    air       2.2
 4 starling  50    air       0.05
 5 cheetah   70    land     50
 6 horse     50    land    450
 7 hippo     30    land   2500
 8 man       25    land     80
 9 squirrel  12    land      0.6
10 cat       35    land      4
11 shark     40    water   600
12 seahorse   0.02 water     0.1
```

```
data |>
  ggplot(aes(x = medium, y = speed, fill = medium)) +
  geom_boxplot(color = "white") + # Set the color of the boxplot whiskers to white
  scale_fill_manual(values = c("air" = "#66c2a5", "land" = "#fc8d62", "water" = "#8da
  labs(title = "Speed by Medium", x = "Medium", y = "Speed") +
  dark_theme_custom
```




```
# Create a new tibble with the count of entries for each medium
medium_counts <- data |>
```

```
  count(medium, name = "count")

# View the new tibble
print(medium_counts)
```

```
# A tibble: 3 × 2
  medium count
  <fct>  <int>
1 air        4
2 land       6
3 water      2
```

Task 2

```
skewed <- function(x_mean, p_50, s) {
  # x_mean - mean value of distribution
  # p_50 - mediam value of distribution
  # s - standard deviation of distribution

  s_index = 3*(x_mean - p_50)*s
  if (s_index > 1) {
    return(c(s_index, "ss")) # Significantly Skewed
  }
  else {
    return(c(s_index, "nss")) # Not Significantly Skewed
  }
}
```

```
# Set the seed for reproducibility
set.seed(123)

# Draw 100 normally distributed values
values <- rnorm(100)

# View the values
print(values)
```

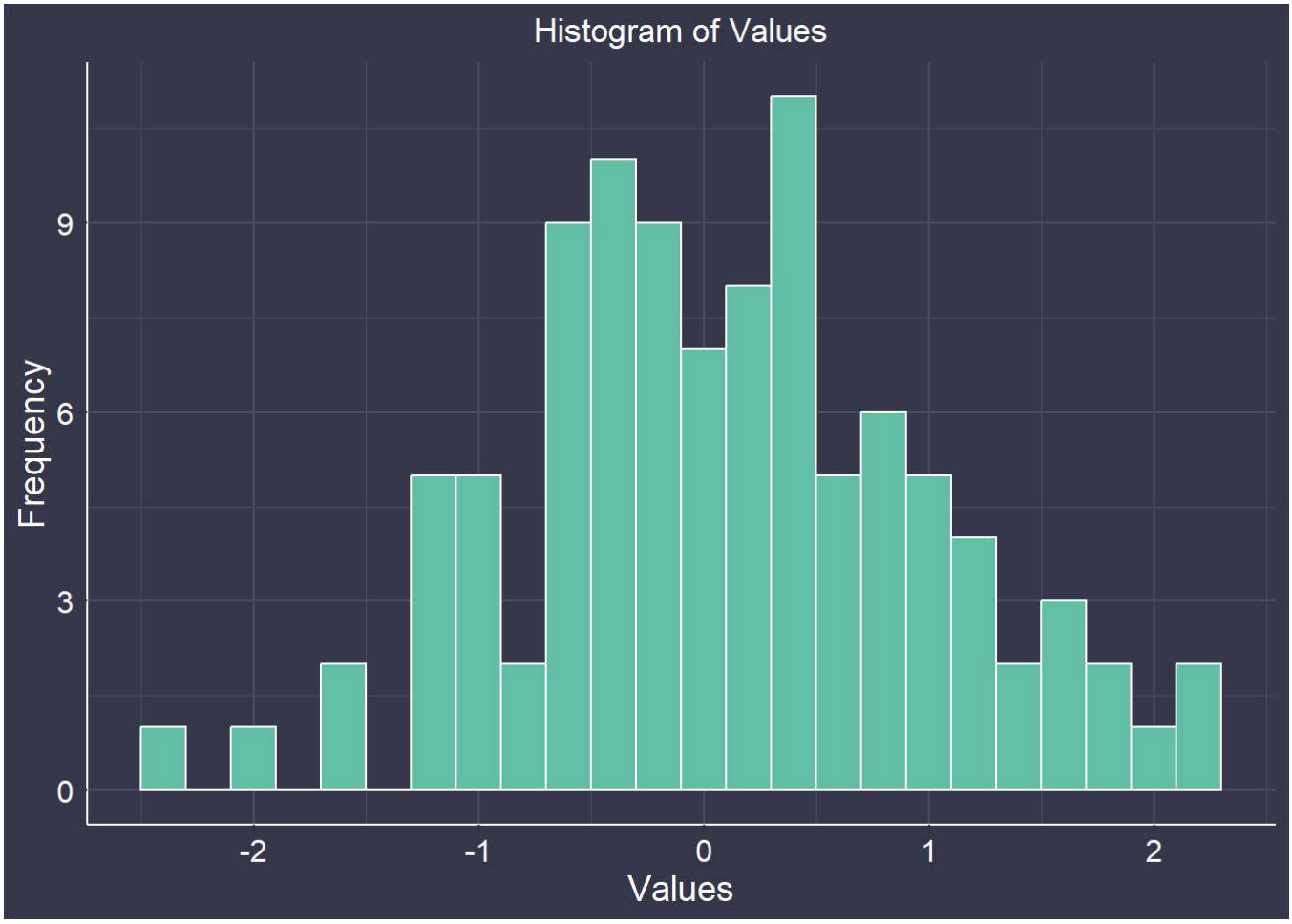
```
  [1] -0.560475647 -0.230177489  1.558708314  0.070508391  0.129287735
  [6]  1.715064987  0.460916206 -1.265061235 -0.686852852 -0.445661970
 [11]  1.224081797  0.359813827  0.400771451  0.110682716 -0.555841135
 [16]  1.786913137  0.497850478 -1.966617157  0.701355902 -0.472791408
 [21] -1.067823706 -0.217974915 -1.026004448 -0.728891229 -0.625039268
 [26] -1.686693311  0.837787044  0.153373118 -1.138136937  1.253814921
 [31]  0.426464221 -0.295071483  0.895125661  0.878133488  0.821581082
 [36]  0.688640254  0.553917654 -0.061911711 -0.305962664 -0.380471001
 [41] -0.694706979 -0.207917278 -1.265396352  2.168955965  1.207961998
 [46] -1.123108583 -0.402884835 -0.466655354  0.779965118 -0.083369066
 [51]  0.253318514 -0.028546755 -0.042870457  1.368602284 -0.225770986
 [56]  1.516470604 -1.548752804  0.584613750  0.123854244  0.215941569
 [61]  0.379639483 -0.502323453 -0.333207384 -1.018575383 -1.071791226
 [66]  0.303528641  0.448209779  0.053004227  0.922267468  2.050084686
```

```
 [71] -0.491031166 -2.309168876  1.005738524 -0.709200763 -0.688008616
 [76]  1.025571370 -0.284773007 -1.220717712  0.181303480 -0.138891362
 [81]  0.005764186  0.385280401 -0.370660032  0.644376549 -0.220486562
 [86]  0.331781964  1.096839013  0.435181491 -0.325931586  1.148807618
 [91]  0.993503856  0.548396960  0.238731735 -0.627906076  1.360652449
 [96] -0.600259587  2.187332993  1.532610626 -0.235700359 -1.026420900
```

```r
# Caculate Skewness Index
result_b <- skewed(mean(values), median(values), sd(values))
print(result_b)
```

```
[1] "0.0784554282709559" "nss"
```

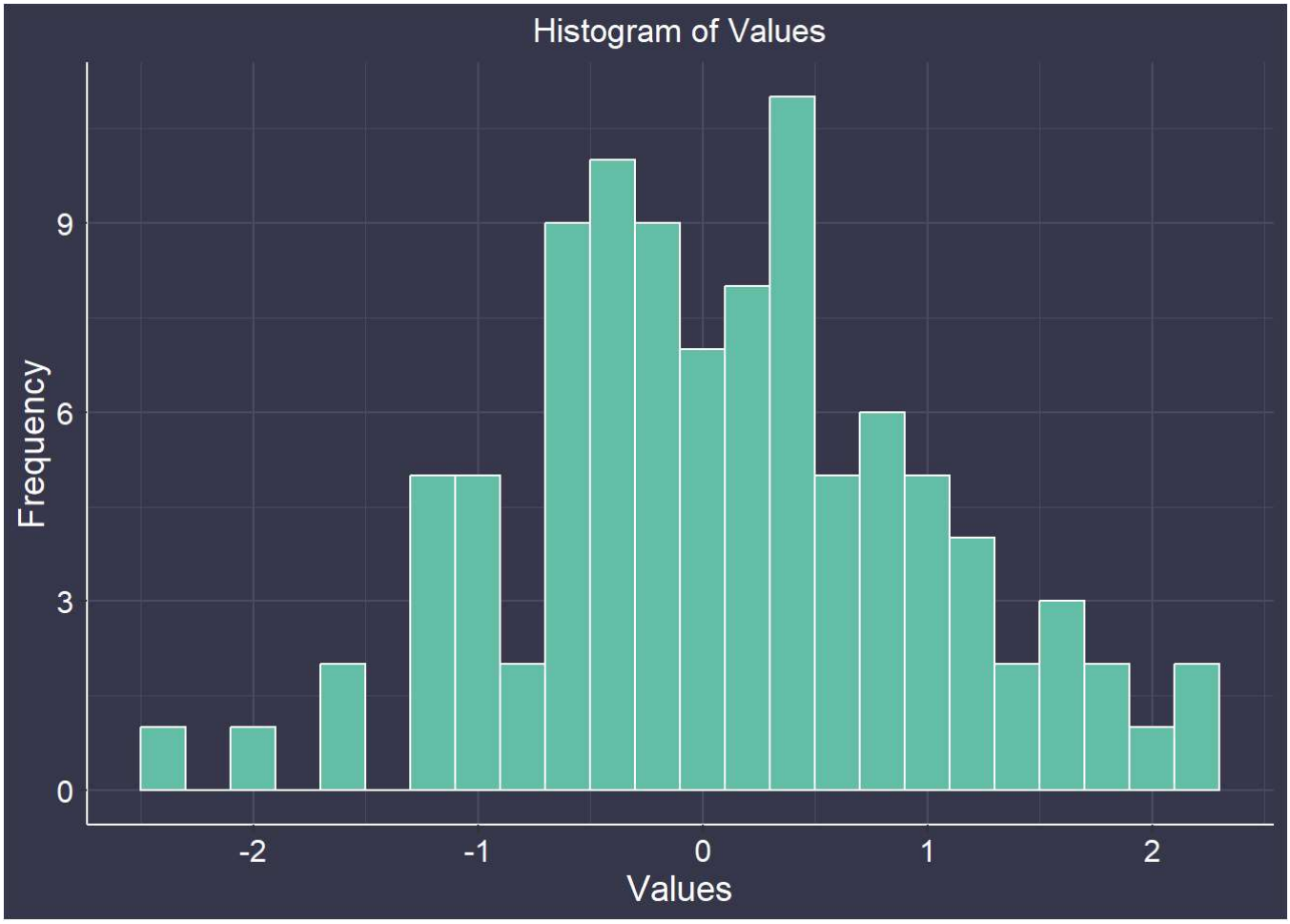
```r
# Plot values
ggplot(data.frame(values), aes(x = values)) +
  geom_histogram(binwidth = 0.2, fill = "#66c2a5", color = "white") +
  labs(title = "Histogram of Values", x = "Values", y = "Frequency") +
  dark_theme_custom
```




```r
values_sq <- values^2
result_c <- skewed(mean(values_sq), median(values_sq), sd(values_sq))
print(result_c)
```

```
[1] "1.73293327593387" "ss"
```

```r
# Plot values
ggplot(data.frame(values_sq), aes(x = values)) +
  geom_histogram(binwidth = 0.2, fill = "#66c2a5", color = "white") +
  labs(title = "Histogram of Values", x = "Values", y = "Frequency") +
  dark_theme_custom
```



Task 3

```r
C_to_F <- function(temp) {
  return ((temp)*9/5) + 32
}

F_to_C <- function(temp) {
  return (temp - 32) * 5/9
}
```