Lab 4

Tian Walker

YYYY-MM-DD

Getting Started

You can download the `transit_cost.csv` data from the website.

```
require(tidyverse)
require(lubridate)
require(ungeviz)
require(dviz.supp)
require(ggtext)
require(countrycode)
require(rio)
require(here)

transit_cost <- import(here("data",'./transit_cost.csv' ))
```

Question 1

Use the transit costs data to reproduce the following plot. To do so, you will need to do a small amount of data cleaning, then calculate the means and standard errors (of the mean) for each country. Please filter to only counties with at least three observations. To use actual country names, rather than abbreviations, join your dataset with the output from the following

```
country_codes <- countrycode::codelist %>%
  select(country_name = country.name.en, country = ecb)
```

Question 2

A local news source reported on Nov 3, 2022, that the percentage of voters supporting Measure 114 was 46.1%. This estimate was based on only 577 voters; therefore, it has a margin of error of 4.1%.

Assume that the margin of error represents twice the standard error of the percentage estimate. Based on this information, create a quantile dot plot to represent the probability that Measure 114 would pass (more than 50% of all voters would support it).

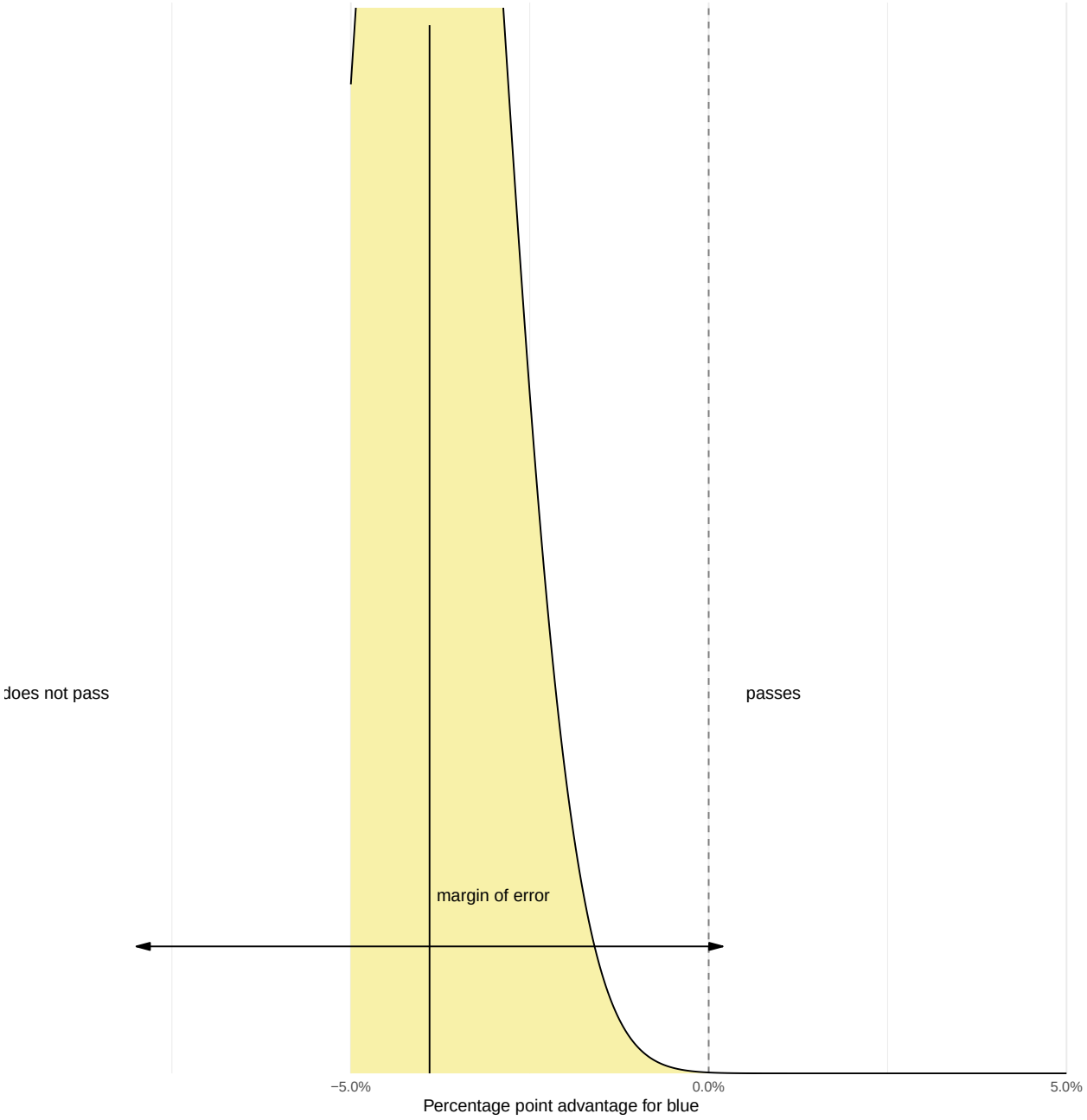
Source: Oregon gun control Measure 114 polls closely

```
mu  <- -3.9
std <- 2.09 # margin of error 4.1 = 1.96*SD,

 x <- seq(-5,5,.01)
```

```
 df_norm <- data.frame(x,
                       y = dnorm(x, mean=mu, sd = ),
                       type = ifelse(x<=0,'Yellow','Blue'))
```

```
  ci_x <- c(qnorm(.025, mu, std), qnorm(0.975, mu, std))
  ci_y <- dnorm(ci_x, mu, std)
```

```
df_annot <- data.frame(
    x = c(mu + 0.05, mu + 0.1, mu + 2.3*std, mu - 2.5*std),
    y = c(dnorm(mu, mu, std) + 0.04, ci_y[1] + 0.01, 3*ci_y[1], 3*ci_y[1]),
    hjust = c(0, 0, 0.5, 0.5),
    vjust = c(-3.14, 0, 0.5, 0.5),
    label = c("best estimate", "margin of error", "passes", "does not pass")
  )

  ggplot(df_norm, aes(x, y)) +
    geom_area(aes(fill = type)) +
    geom_vline(xintercept = 0, linetype = 2, color = "gray50") +
    geom_line() +
    geom_segment(
      data = data.frame(x = 1),
      x = ci_x[1], xend = ci_x[2], y = ci_y[1], yend = ci_y[2],
      arrow = arrow(angle = 15,
                    length = grid::unit(9, "pt"),
                    ends = "both",
                    type = "closed"),
      inherit.aes = FALSE
    ) +
    geom_segment(
      data = data.frame(x = 1),
      x = mu, xend = mu, y = 0, yend = dnorm(mu, mu, std) + 0.04,
      inherit.aes = FALSE
    ) +
    geom_text(
      data = df_annot,
      aes(x, y, label = label, hjust = hjust, vjust = vjust)) +
    scale_x_continuous(
      name = "Percentage point advantage for blue",
      labels = scales::percent_format(accuracy = 0.1, scale = 1)
    ) +
    scale_y_continuous(
      name = NULL,
      breaks = NULL,
      expand = c(0, 0),
      limits = c(0, dnorm(mu, mu, std) + 0.045)
    ) +
    scale_fill_manual(
      values = c(Yellow = "#f8f1a9", Blue = "#b1daf4"),
      guide = "none"
    ) +
    theme_minimal()
```




```
    pnorm(0,mu,std,lower.tail = TRUE)
```

```
## [1] 0.9689813
```

```
    pnorm(0,mu,std,lower.tail = FALSE)
```

```
## [1] 0.03101867
```

```
    # The chance of the yellow party winning is 13.3%.
    # The chance of the yellow party winning is 87.7%.
```

```
##############################################################################
#
#  3. Quantile dotplot = Discrete outcome + Continuous Distribution
```

```
#
################################################################################

# Parameters

mu  <- 1
std <- 0.9 # margin of error 1.76 = 1.96*SD,

x <- seq(0.01,.99,.01) # generate a sequence of probabilities
                       # from 1% to 99% by 1% increments

# Quantile data frame based on normal distribution

  q_df <- data.frame(x = x,
                     q = qnorm(x, mu, std))

  q_df$winner <- ifelse(q_df$q <= 0,"#f8f1a9","#b1daf4")

  q_df
```

```
##       x           q  winner
## 1  0.01 -1.09371309 #f8f1a9
## 2  0.02 -0.84837402 #f8f1a9
## 3  0.03 -0.69271425 #f8f1a9
## 4  0.04 -0.57561746 #f8f1a9
## 5  0.05 -0.48036826 #f8f1a9
## 6  0.06 -0.39929624 #f8f1a9
## 7  0.07 -0.32821193 #f8f1a9
## 8  0.08 -0.26456440 #f8f1a9
## 9  0.09 -0.20667953 #f8f1a9
## 10 0.10 -0.15339641 #f8f1a9
## 11 0.11 -0.10387531 #f8f1a9
## 12 0.12 -0.05748811 #f8f1a9
## 13 0.13 -0.01375202 #f8f1a9
## 14 0.14  0.02771259 #b1daf4
## 15 0.15  0.06720995 #b1daf4
## 16 0.16  0.10498791 #b1daf4
## 17 0.17  0.14125127 #b1daf4
## 18 0.18  0.17617142 #b1daf4
## 19 0.19  0.20989333 #b1daf4
## 20 0.20  0.24254089 #b1daf4
## 21 0.21  0.27422088 #b1daf4
## 22 0.22  0.30502611 #b1daf4
## 23 0.23  0.33503784 #b1daf4
## 24 0.24  0.36432769 #b1daf4
## 25 0.25  0.39295922 #b1daf4
## 26 0.26  0.42098914 #b1daf4
## 27 0.27  0.44846831 #b1daf4
## 28 0.28  0.47544264 #b1daf4
## 29 0.29  0.50195375 #b1daf4
## 30 0.30  0.52803954 #b1daf4
## 31 0.31  0.55373469 #b1daf4
## 32 0.32  0.57907108 #b1daf4
## 33 0.33  0.60407815 #b1daf4
```

```
## 34 0.34  0.62878318 #b1daf4
## 35 0.35  0.65321158 #b1daf4
## 36 0.36  0.67738709 #b1daf4
## 37 0.37  0.70133199 #b1daf4
## 38 0.38  0.72506729 #b1daf4
## 39 0.39  0.74861287 #b1daf4
## 40 0.40  0.77198761 #b1daf4
## 41 0.41  0.79520952 #b1daf4
## 42 0.42  0.81829587 #b1daf4
## 43 0.43  0.84126325 #b1daf4
## 44 0.44  0.86412771 #b1daf4
## 45 0.45  0.88690479 #b1daf4
## 46 0.46  0.90960965 #b1daf4
## 47 0.47  0.93225712 #b1daf4
## 48 0.48  0.95486177 #b1daf4
## 49 0.49  0.97743798 #b1daf4
## 50 0.50  1.00000000 #b1daf4
## 51 0.51  1.02256202 #b1daf4
## 52 0.52  1.04513823 #b1daf4
## 53 0.53  1.06774288 #b1daf4
## 54 0.54  1.09039035 #b1daf4
## 55 0.55  1.11309521 #b1daf4
## 56 0.56  1.13587229 #b1daf4
## 57 0.57  1.15873675 #b1daf4
## 58 0.58  1.18170413 #b1daf4
## 59 0.59  1.20479048 #b1daf4
## 60 0.60  1.22801239 #b1daf4
## 61 0.61  1.25138713 #b1daf4
## 62 0.62  1.27493271 #b1daf4
## 63 0.63  1.29866801 #b1daf4
## 64 0.64  1.32261291 #b1daf4
## 65 0.65  1.34678842 #b1daf4
## 66 0.66  1.37121682 #b1daf4
## 67 0.67  1.39592185 #b1daf4
## 68 0.68  1.42092892 #b1daf4
## 69 0.69  1.44626531 #b1daf4
## 70 0.70  1.47196046 #b1daf4
## 71 0.71  1.49804625 #b1daf4
## 72 0.72  1.52455736 #b1daf4
## 73 0.73  1.55153169 #b1daf4
## 74 0.74  1.57901086 #b1daf4
## 75 0.75  1.60704078 #b1daf4
## 76 0.76  1.63567231 #b1daf4
## 77 0.77  1.66496216 #b1daf4
## 78 0.78  1.69497389 #b1daf4
## 79 0.79  1.72577912 #b1daf4
## 80 0.80  1.75745911 #b1daf4
## 81 0.81  1.79010667 #b1daf4
## 82 0.82  1.82382858 #b1daf4
## 83 0.83  1.85874873 #b1daf4
## 84 0.84  1.89501209 #b1daf4
## 85 0.85  1.93279005 #b1daf4
## 86 0.86  1.97228741 #b1daf4
## 87 0.87  2.01375202 #b1daf4
```

```
## 88 0.88  2.05748811 #b1daf4
## 89 0.89  2.10387531 #b1daf4
## 90 0.90  2.15339641 #b1daf4
## 91 0.91  2.20667953 #b1daf4
## 92 0.92  2.26456440 #b1daf4
## 93 0.93  2.32821193 #b1daf4
## 94 0.94  2.39929624 #b1daf4
## 95 0.95  2.48036826 #b1daf4
## 96 0.96  2.57561746 #b1daf4
## 97 0.97  2.69271425 #b1daf4
## 98 0.98  2.84837402 #b1daf4
## 99 0.99  3.09371309 #b1daf4
```