Lab 4

Seulbi Lee

20 February, 2023

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(plotrix)

transit_cost <- read_csv('./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)

transit_cost <- left_join(transit_cost, country_codes) |> drop_na()
transit_cost[is.na(transit_cost)] <- 0

str(transit_cost)
```

```
## tibble [426 x 21] (S3: tbl_df/tbl/data.frame)
##  $ e               : num [1:426] 7136 7137 7138 7139 7144 ...
##  $ country         : chr [1:426] "CA" "CA" "CA" "CA" ...
##  $ city            : chr [1:426] "Vancouver" "Toronto" "Toronto" "Toronto" ...
##  $ line            : chr [1:426] "Broadway" "Vaughan" "Scarborough" "Ontario" ...
##  $ start_year      : chr [1:426] "2020" "2009" "2020" "2020" ...
##  $ end_year        : chr [1:426] "2025" "2017" "2030" "2030" ...
##  $ rr              : num [1:426] 0 0 0 0 0 0 0 0 0 0 ...
##  $ length          : num [1:426] 5.7 8.6 7.8 15.5 7.4 9.7 5.8 5.1 4.2 4.2 ...
##  $ tunnel_per      : chr [1:426] "87.72%" "100.00%" "100.00%" "57.00%" ...
##  $ tunnel          : num [1:426] 5 8.6 7.8 8.8 7.4 7.1 5.8 5.1 4.2 4.2 ...
```

```
##  $ stations        : num [1:426] 6 6 3 15 6 8 5 2 2 2 ...
##  $ source1         : chr [1:426] "Plan" "Media" "Wiki" "Plan" ...
##  $ cost            : num [1:426] 2830 3200 5500 8573 5600 ...
##  $ currency        : chr [1:426] "CAD" "CAD" "CAD" "CAD" ...
##  $ year            : num [1:426] 2018 2013 2018 2019 2020 ...
##  $ ppp_rate        : num [1:426] 0.84 0.81 0.84 0.84 0.84 1.3 0.84 1 1 1 ...
##  $ real_cost       : chr [1:426] "2377.2" "2592" "4620" "7201.32" ...
##  $ cost_km_millions: num [1:426] 417 301 592 465 636 ...
##  $ source2         : chr [1:426] "Media" "Media" "Media" "Plan" ...
##  $ reference       : chr [1:426] "https://www.translink.ca/Plans-and-Projects/Rapid-Transit-Projects/
##  $ country_name    : chr [1:426] "Canada" "Canada" "Canada" "Canada" ...
```

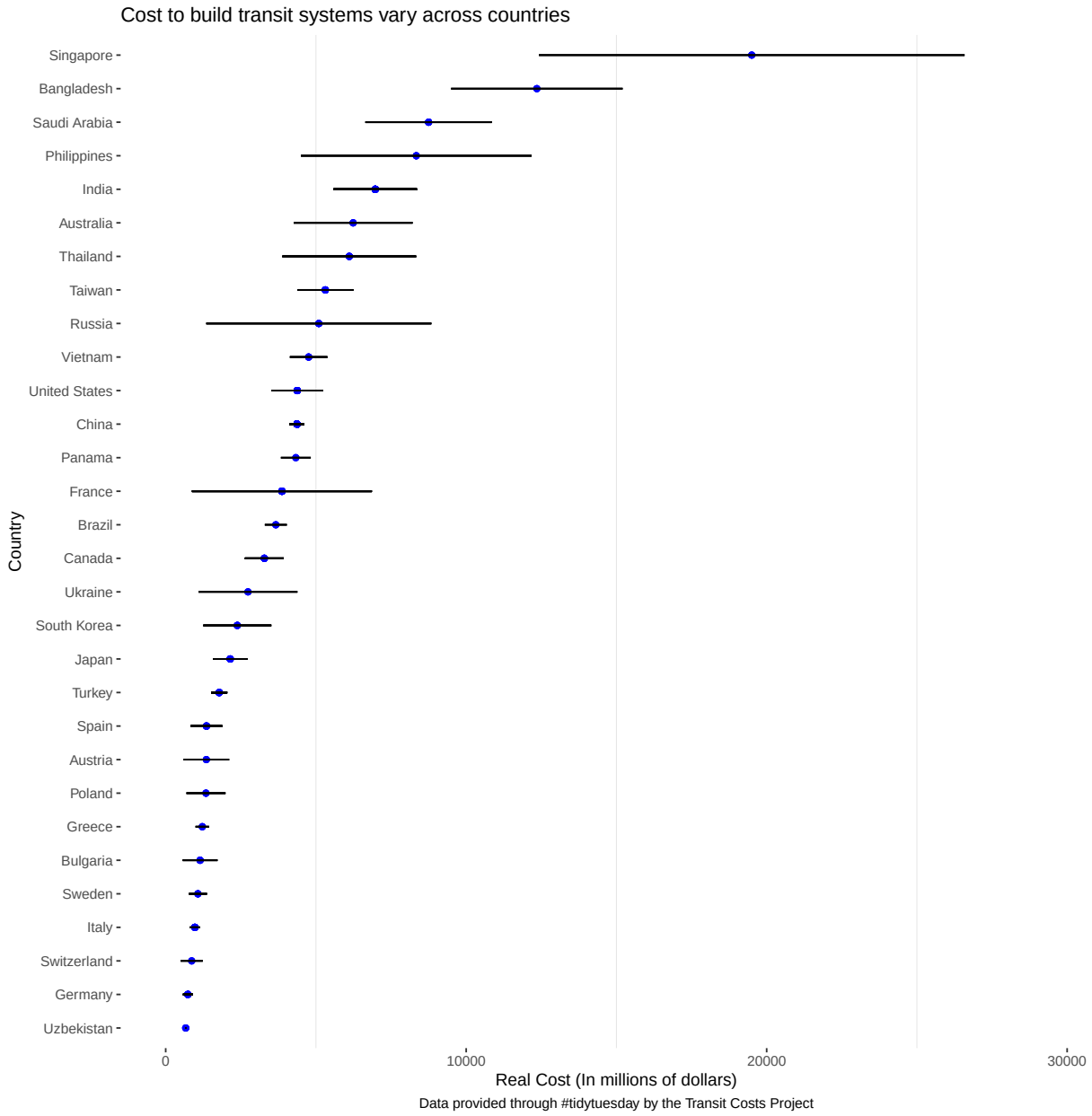
```
transit_cost$real_cost <- as.numeric(transit_cost$real_cost)


transit_mean <- aggregate(transit_cost$real_cost, list(transit_cost$country), FUN=mean)
colnames(transit_mean)[colnames(transit_mean) == "Group.1"] ="country"
colnames(transit_mean)[colnames(transit_mean) == "x"] ="mean"

transit_error <- aggregate(transit_cost$real_cost, list(transit_cost$country), FUN=std.error)
colnames(transit_error)[colnames(transit_error) == "Group.1"] ="country"
colnames(transit_error)[colnames(transit_error) == "x"] ="std.error"

transit_new <- left_join(transit_mean, transit_error)

dat <- left_join(transit_cost, transit_new)

table(dat$country)
```

```
##
##  AE  AR  AT  AU  BD  BE  BG  BR  CA  CH  CL  CN  CZ  DE  DK  EC  EG  ES  FI  FR
##   1   1   3   4   3   1   5   3  10   3   2 169   1  12   1   1   2  15   2  15
##  GR  HU  ID  IL  IN  IR  IT  JP  KR  KW  MX  MY  NL  NO  NZ  PA  PE  PH  PK  PL
##   4   1   1   2  27   1  11  14   6   1   2   2   1   2   1   3   1   4   1   4
##  PT  RO  RU  SA  SE  SG  TH  TR  TW  UA  US  UZ  VN
##   2   1   5   6   5   3   7  18  11   3  13   4   5
```

```
dat <- subset(dat, country %in% c('AT','AU','BD','BG','BR','CA','CH','CN','DE','ES','FR','GR','IN','IT'
  arrange(desc(mean))
```

```
dat |>
  ggplot(aes(x=stats::reorder(country_name, mean), y= mean)) +
    geom_point(color="blue")+
    geom_linerange(aes(ymin=mean-std.error,
                       ymax=mean+std.error),
                   width=0.5, size=0.5)+
    coord_flip()+
    labs(title="Cost to build transit systems vary across countries",
         x="Country", y="Real Cost (In millions of dollars)",
         caption="Data provided through #tidytuesday by the Transit Costs Project") +
    theme(legend.position='none',
          panel.grid.major = element_blank(),
          panel.grid.minor = element_line(colour = "grey90"),
```

```
        panel.background = element_blank(),
        plot.caption = element_text(hjust=0.5)) +
ylim(0, 30000)
```



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%.

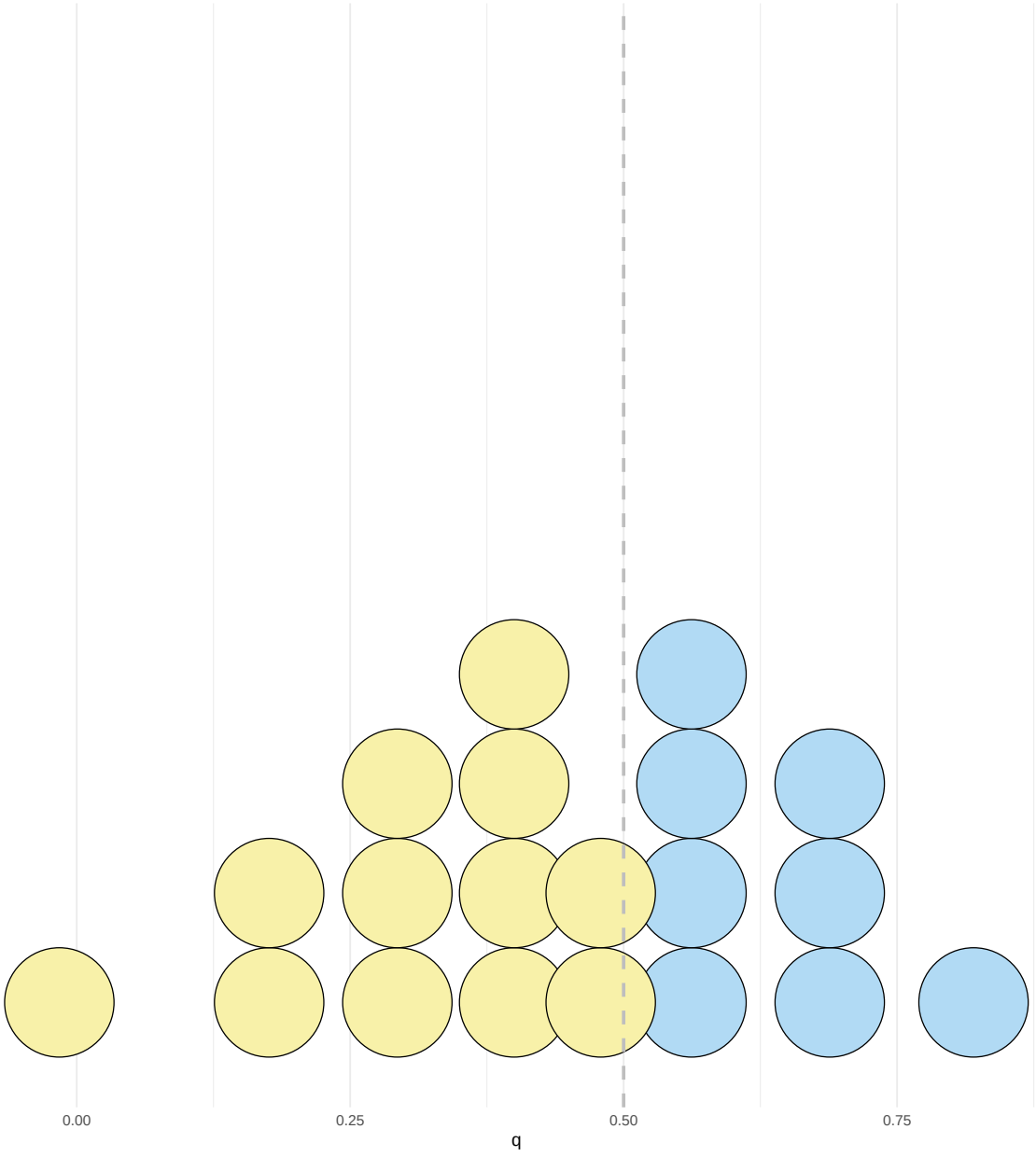
Assume that the margin of error represents twice the standard error of the percentage estimate. Based on this information, create a quantile dot plot to represent the probability that Measure 114 would pass (more than 50% of all voters would support it).

Source: Oregon gun control Measure 114 polls closely

```r
mu  <- 0.461
std <- 0.41/2 # margin of error 0.41 = 2 * std

x <- seq(0.01,.99,.05) # 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 < .5,"#f8f1a9","#b1daf4")

  q_df
```

```
##       x           q  winner
## 1  0.01 -0.01590131 #f8f1a9
## 2  0.06  0.14227141 #f8f1a9
## 3  0.11  0.20956174 #f8f1a9
## 4  0.16  0.25713613 #f8f1a9
## 5  0.21  0.29568364 #f8f1a9
## 6  0.26  0.32911419 #f8f1a9
## 7  0.31  0.35935068 #f8f1a9
## 8  0.36  0.38751595 #f8f1a9
## 9  0.41  0.41435328 #f8f1a9
## 10 0.46  0.44041109 #f8f1a9
## 11 0.51  0.46613913 #f8f1a9
## 12 0.56  0.49194869 #f8f1a9
## 13 0.61  0.51826040 #b1daf4
## 14 0.66  0.54555494 #b1daf4
## 15 0.71  0.57444387 #b1daf4
## 16 0.76  0.60579203 #b1daf4
## 17 0.81  0.64096874 #b1daf4
## 18 0.86  0.68246546 #b1daf4
## 19 0.91  0.73585478 #b1daf4
## 20 0.96  0.81989064 #b1daf4
```

```r
  ggplot(q_df, aes(q)) +
    geom_dotplot(aes(fill=winner), binwidth = 0.1)+
    theme_minimal()+
    geom_vline(xintercept = 0.5,
               color = "gray",
               linetype = "dashed",
               size = 1)+
    scale_fill_identity(guide='none')+
    scale_y_continuous(name = "",
                       breaks = NULL)
```


0.00
0.25
0.50
0.75
q