Lab 4

Lucy Whitmore

2023-02-14

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)

transit_cost <- read_csv('./transit_cost.csv')
```

Question 1

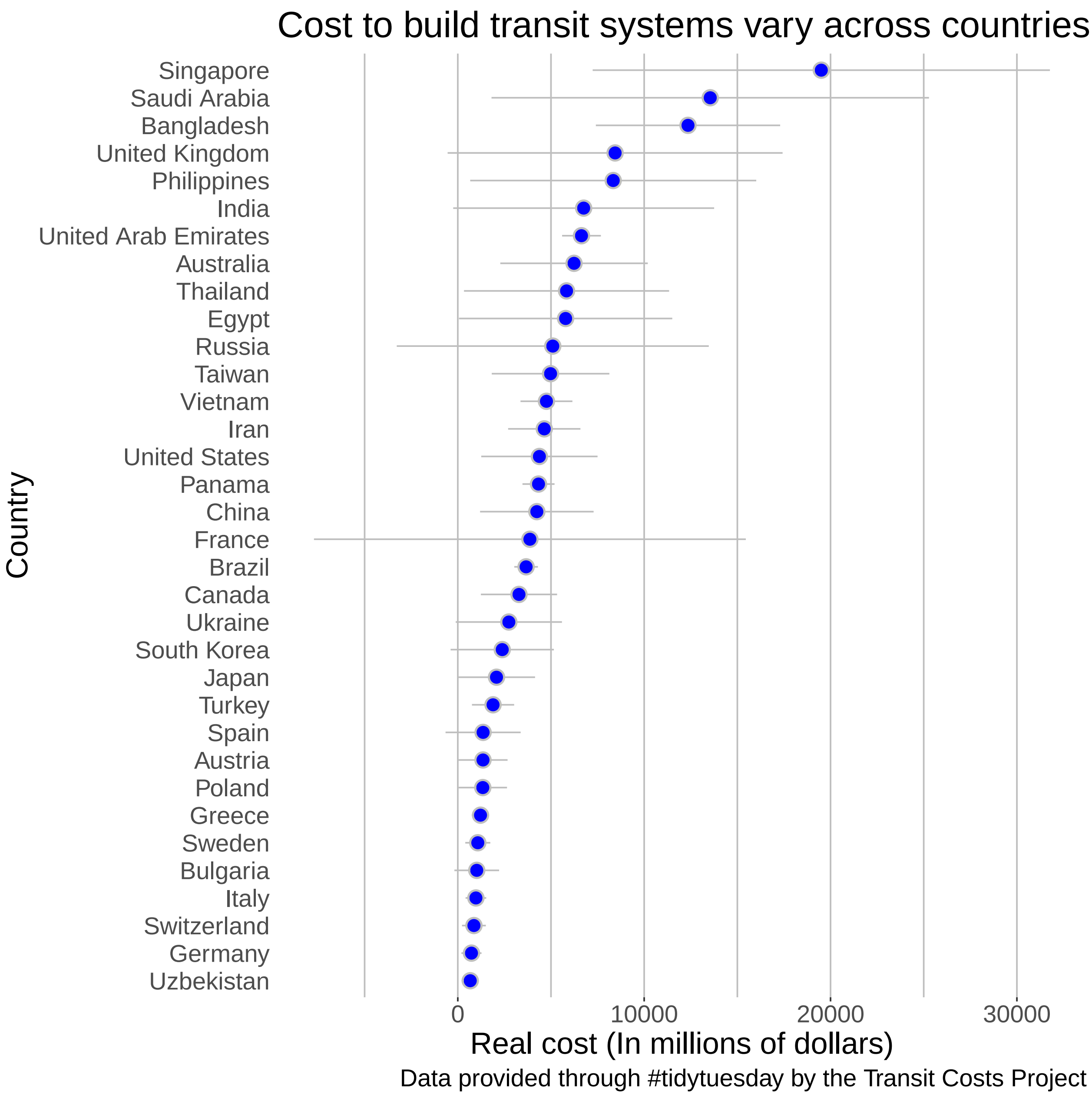
Use the transit costs data to reproduce the following plot. To do so, you will need to do a small amount of data cleaning, then calculate the means and standard errors (of the mean) for each country. Please filter to only counties with at least three observations. To use actual country names, rather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

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

```
# calculate means and standard deviations, count observations
transit_cost_means <- transit_cost %>%
  group_by(country) %>%
  mutate(real_cost = as.numeric(real_cost)) %>%
  summarise(m_cost=mean(real_cost, na.rm=T), sd_cost = sd(real_cost, na.rm=T), n = n()) %>%
  ungroup() %>%
# filter to countries with >2 observations
  filter(n>2) %>%
  na.omit() %>%
  arrange(m_cost)

#rename UK to GB so that datasets will merge properly
transit_cost_means["country"][transit_cost_means["country"] == "UK"] <- "GB"

# join datasets to get country names
transit_countries <- left_join(transit_cost_means, country_codes)

transit_countries$country_name <- factor(transit_countries$country_name, levels = transit_countries$coun
```

```
p <- transit_countries %>%
  ggplot(aes(x=m_cost, y=country_name, xmin=m_cost-sd_cost, xmax=m_cost+sd_cost))
```

```r
p +  geom_pointrange(fill='blue', color='grey', shape=21, size=1) +
  theme(
    plot.background = element_rect(fill = "white"),
    panel.background = element_rect(fill = "white"),
    panel.grid.major = element_line(size = 0.5, linetype = 'solid',
                                    colour = "grey"),
    panel.grid.minor = element_line(size = 0.5, linetype = 'solid',
                                    colour = "grey"),
    panel.grid.major.y = element_blank(),
    panel.grid.minor.y = element_blank(),
    axis.ticks.y=element_blank(),
   text = element_text(size = 20)
    ) +
  labs(x="Real cost (In millions of dollars)", y = "Country",title="Cost to build transit systems vary a
```



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

```
# Parameters

mu  <- 3.9 #advantage: 50-46.1 = 3.9
std <- 2.09 # margin of error 4.1     , 1.76 = 1.96*SD,
```

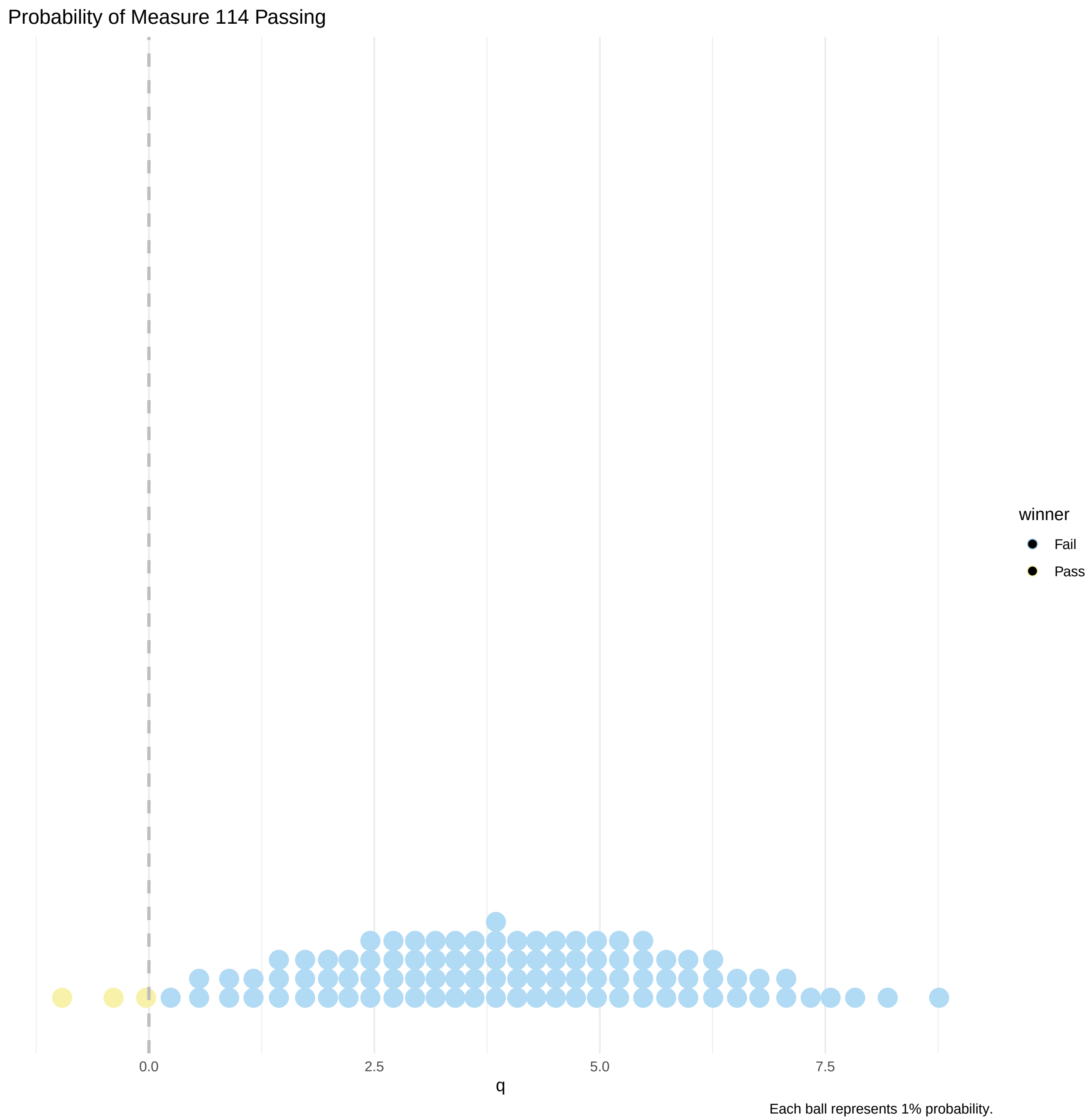
```r
# margin of error is 4.1
#example: 4.1 = 1.96*SD

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 -0.96206706 #f8f1a9
## 2  0.02 -0.39233522 #f8f1a9
## 3  0.03 -0.03085864 #f8f1a9
## 4  0.04  0.24106611 #b1daf4
## 5  0.05  0.46225592 #b1daf4
## 6  0.06  0.65052319 #b1daf4
## 7  0.07  0.81559675 #b1daf4
## 8  0.08  0.96340044 #b1daf4
## 9  0.09  1.09782198 #b1daf4
## 10 0.10  1.22155723 #b1daf4
## 11 0.11  1.33655623 #b1daf4
## 12 0.12  1.44427760 #b1daf4
## 13 0.13  1.54584254 #b1daf4
## 14 0.14  1.64213258 #b1daf4
## 15 0.15  1.73385422 #b1daf4
## 16 0.16  1.82158302 #b1daf4
## 17 0.17  1.90579462 #b1daf4
## 18 0.18  1.98688697 #b1daf4
## 19 0.19  2.06519674 #b1daf4
## 20 0.20  2.14101162 #b1daf4
## 21 0.21  2.21457959 #b1daf4
## 22 0.22  2.28611618 #b1daf4
## 23 0.23  2.35581009 #b1daf4
## 24 0.24  2.42382764 #b1daf4
## 25 0.25  2.49031642 #b1daf4
## 26 0.26  2.55540810 #b1daf4
## 27 0.27  2.61922085 #b1daf4
## 28 0.28  2.68186125 #b1daf4
## 29 0.29  2.74342594 #b1daf4
## 30 0.30  2.80400293 #b1daf4
## 31 0.31  2.86367277 #b1daf4
## 32 0.32  2.92250951 #b1daf4
## 33 0.33  2.98058148 #b1daf4
## 34 0.34  3.03795206 #b1daf4
## 35 0.35  3.09468023 #b1daf4
## 36 0.36  3.15082112 #b1daf4
## 37 0.37  3.20642651 #b1daf4
```

```
## 38 0.38  3.26154515 #b1daf4
## 39 0.39  3.31622322 #b1daf4
## 40 0.40  3.37050455 #b1daf4
## 41 0.41  3.42443100 #b1daf4
## 42 0.42  3.47804263 #b1daf4
## 43 0.43  3.53137800 #b1daf4
## 44 0.44  3.58447434 #b1daf4
## 45 0.45  3.63736779 #b1daf4
## 46 0.46  3.69009352 #b1daf4
## 47 0.47  3.74268599 #b1daf4
## 48 0.48  3.79517901 #b1daf4
## 49 0.49  3.84760598 #b1daf4
## 50 0.50  3.90000000 #b1daf4
## 51 0.51  3.95239402 #b1daf4
## 52 0.52  4.00482099 #b1daf4
## 53 0.53  4.05731401 #b1daf4
## 54 0.54  4.10990648 #b1daf4
## 55 0.55  4.16263221 #b1daf4
## 56 0.56  4.21552566 #b1daf4
## 57 0.57  4.26862200 #b1daf4
## 58 0.58  4.32195737 #b1daf4
## 59 0.59  4.37556900 #b1daf4
## 60 0.60  4.42949545 #b1daf4
## 61 0.61  4.48377678 #b1daf4
## 62 0.62  4.53845485 #b1daf4
## 63 0.63  4.59357349 #b1daf4
## 64 0.64  4.64917888 #b1daf4
## 65 0.65  4.70531977 #b1daf4
## 66 0.66  4.76204794 #b1daf4
## 67 0.67  4.81941852 #b1daf4
## 68 0.68  4.87749049 #b1daf4
## 69 0.69  4.93632723 #b1daf4
## 70 0.70  4.99599707 #b1daf4
## 71 0.71  5.05657406 #b1daf4
## 72 0.72  5.11813875 #b1daf4
## 73 0.73  5.18077915 #b1daf4
## 74 0.74  5.24459190 #b1daf4
## 75 0.75  5.30968358 #b1daf4
## 76 0.76  5.37617236 #b1daf4
## 77 0.77  5.44418991 #b1daf4
## 78 0.78  5.51388382 #b1daf4
## 79 0.79  5.58542041 #b1daf4
## 80 0.80  5.65898838 #b1daf4
## 81 0.81  5.73480326 #b1daf4
## 82 0.82  5.81311303 #b1daf4
## 83 0.83  5.89420538 #b1daf4
## 84 0.84  5.97841698 #b1daf4
## 85 0.85  6.06614578 #b1daf4
## 86 0.86  6.15786742 #b1daf4
## 87 0.87  6.25415746 #b1daf4
## 88 0.88  6.35572240 #b1daf4
## 89 0.89  6.46344377 #b1daf4
## 90 0.90  6.57844277 #b1daf4
## 91 0.91  6.70217802 #b1daf4
```

```
## 92 0.92  6.83659956 #b1daf4
## 93 0.93  6.98440325 #b1daf4
## 94 0.94  7.14947681 #b1daf4
## 95 0.95  7.33774408 #b1daf4
## 96 0.96  7.55893389 #b1daf4
## 97 0.97  7.83085864 #b1daf4
## 98 0.98  8.19233522 #b1daf4
## 99 0.99  8.76206706 #b1daf4
```

```
# Dotplot

  ggplot(q_df, aes(q)) +
    geom_dotplot(aes(color=winner, fill=winner),binwidth=0.21)+
    theme_minimal()+
    scale_color_manual(labels = c("Fail", "Pass"), values = c("#b1daf4", "#f8f1a9")) +
    geom_vline(xintercept = 0,
               color = "gray",
               linetype = "dashed",
               size = 1)+
    scale_fill_identity(guide='none')+
    scale_y_continuous(name = "",
                       breaks = NULL) +
    labs(title = "Probability of Measure 114 Passing", caption = "Each ball represents 1% probability.")
```


Probability of Measure 114 Passing
winner
Fail
Pass
0.0
2.5
5.0
7.5
q
Each ball represents 1% probability.