Code for QSS tidyverse Chapter 3: Measurement

Kosuke Imai and Nora Webb Williams

First Printing

Measurement

Measuring Civilian Victimization during Wartime

```
library(tidyverse)
library(qss)

## Load in the data from QSS package
data(afghan, package = "qss")

## Summarize the main variables
afghan %>%
  select(age, educ.years, employed, income) %>%
  summary()
```

```
##       age          educ.years        employed
##  Min.   :15.00   Min.   : 0.000   Min.   :0.0000
##  1st Qu.:22.00   1st Qu.: 0.000   1st Qu.:0.0000
##  Median :30.00   Median : 1.000   Median :1.0000
##  Mean   :32.39   Mean   : 4.002   Mean   :0.5828
##  3rd Qu.:40.00   3rd Qu.: 8.000   3rd Qu.:1.0000
##  Max.   :80.00   Max.   :18.000   Max.   :1.0000
##     income
##  Length:2754
##  Class :character
##  Mode  :character
##
##
##
```

```
count(afghan, income)
```

```
##           income    n
## 1   10,001-20,000  616
## 2    2,001-10,000 1420
## 3   20,001-30,000   93
## 4 less than 2,000  457
## 5     over 30,000   14
## 6            <NA>  154
```

```r
unique(afghan$income)
```

```
## [1] "2,001-10,000"    NA                 "10,001-20,000"
## [4] "less than 2,000" "20,001-30,000"   "over 30,000"
```

```r
## What proportion of respondents were harmed by
## ISAF and/or Taliban?
harm_props <- afghan %>%
  group_by(violent.exp.ISAF, violent.exp.taliban) %>%
  count() %>%
  ungroup() %>%
  mutate(prop = n / sum(n))

harm_props
```

```
## # A tibble: 9 x 4
##   violent.exp.ISAF violent.exp.taliban     n    prop
##              <int>               <int> <int>   <dbl>
## 1                0                   0  1330 0.483
## 2                0                   1   354 0.129
## 3                0                  NA    22 0.00799
## 4                1                   0   475 0.172
## 5                1                   1   526 0.191
## 6                1                  NA    22 0.00799
## 7               NA                   0     7 0.00254
## 8               NA                   1     8 0.00290
## 9               NA                  NA    10 0.00363
```

```r
## without ungroup(), commenting out the line
afghan %>%
  group_by(violent.exp.ISAF, violent.exp.taliban) %>%
  count() %>%
##  ungroup() %>%
  mutate(prop = n / sum(n))
```

```
## # A tibble: 9 x 4
## # Groups:   violent.exp.ISAF, violent.exp.taliban [9]
##   violent.exp.ISAF violent.exp.taliban     n  prop
##              <int>               <int> <int> <dbl>
## 1                0                   0  1330     1
## 2                0                   1   354     1
## 3                0                  NA    22     1
## 4                1                   0   475     1
## 5                1                   1   526     1
## 6                1                  NA    22     1
## 7               NA                   0     7     1
## 8               NA                   1     8     1
## 9               NA                  NA    10     1
```

```r
## What proportion of respondents were harmed by ISAF?
ISAF_harm_prop <- harm_props %>%
```

```r
  filter(violent.exp.ISAF == 1) %>%
  summarize(harm_prop = sum(prop)) %>%
  pull()

ISAF_harm_prop
```

```
## [1] 0.3714597
```

```r
## What proportion of respondents were harmed by Taliban?
talib_harm_prop <- harm_props %>%
  filter(violent.exp.taliban == 1) %>%
  summarize(harm_prop = sum(prop)) %>%
  pull()

talib_harm_prop
```

```
## [1] 0.3224401
```

```r
## What proportion of respondents were harmed by both?
both_harm_prop <- harm_props %>%
  filter(violent.exp.taliban == 1 &
           violent.exp.ISAF == 1) %>%
  summarize(harm_prop = sum(prop)) %>%
  pull()

both_harm_prop
```

```
## [1] 0.1909949
```

Handling Missing Data in R

```r
## print income data for first 10 respondents
afghan %>%
  select(income) %>%
  slice(1:10)
```

```
##           income
## 1   2,001-10,000
## 2   2,001-10,000
## 3   2,001-10,000
## 4   2,001-10,000
## 5   2,001-10,000
## 6           <NA>
## 7  10,001-20,000
## 8   2,001-10,000
## 9   2,001-10,000
## 10          <NA>
```

```r
## What is.na() returns for these observations
afghan %>%
  select(income) %>%
  slice(1:10) %>%
  is.na()
```

```
##       income
##  [1,]  FALSE
##  [2,]  FALSE
##  [3,]  FALSE
##  [4,]  FALSE
##  [5,]  FALSE
##  [6,]   TRUE
##  [7,]  FALSE
##  [8,]  FALSE
##  [9,]  FALSE
## [10,]   TRUE
```

```r
## What number and proportion of responses are missing for income?
summarize(afghan,
          n_missing = sum(is.na(income)),
          p_missing = mean(is.na(income)))
```

```
##   n_missing  p_missing
## 1       154 0.05591866
```

```r
x <- c(1, 2, 3, NA)
mean(x)
```

```
## [1] NA
```

```r
mean(x, na.rm = TRUE)
```

```
## [1] 2
```

```r
## Table for non-missing values of ISAF and Taliban
afghan %>%
  filter(!is.na(violent.exp.ISAF), !is.na(violent.exp.taliban)) %>%
  group_by(violent.exp.ISAF, violent.exp.taliban) %>%
  count() %>%
  ungroup() %>%
  mutate(prop = n / sum(n)) %>%
  arrange(prop) #compare to arrange(desc(prop))
```

```
## # A tibble: 4 x 4
##   violent.exp.ISAF violent.exp.taliban     n  prop
##              <int>               <int> <int> <dbl>
## 1                0                   1   354 0.132
## 2                1                   0   475 0.177
## 3                1                   1   526 0.196
## 4                0                   0  1330 0.495
```

```r
## Reminder of what harm_props is
harm_props
```

```
## # A tibble: 9 x 4
##   violent.exp.ISAF violent.exp.taliban     n    prop
##              <int>               <int> <int>   <dbl>
## 1                0                   0  1330 0.483
## 2                0                   1   354 0.129
## 3                0                  NA    22 0.00799
## 4                1                   0   475 0.172
## 5                1                   1   526 0.191
## 6                1                  NA    22 0.00799
## 7               NA                   0     7 0.00254
## 8               NA                   1     8 0.00290
## 9               NA                  NA    10 0.00363
```

```r
## What proportion of observations are missing for either
## ISAF or Taliban harm?
missing_prop <- harm_props %>%
  filter(is.na(violent.exp.ISAF) | is.na(violent.exp.taliban)) %>%
  ungroup() %>%
  summarize(missing_prop = sum(prop)) %>%
  pull()

missing_prop
```

```
## [1] 0.02505447
```

```r
afghan.sub <- na.omit(afghan)  # listwise deletion
nrow(afghan.sub)
```

```
## [1] 2554
```

```r
afghan.sub.2 <- drop_na(afghan) # equivalent with drop_na()
nrow(afghan.sub.2)
```

```
## [1] 2554
```

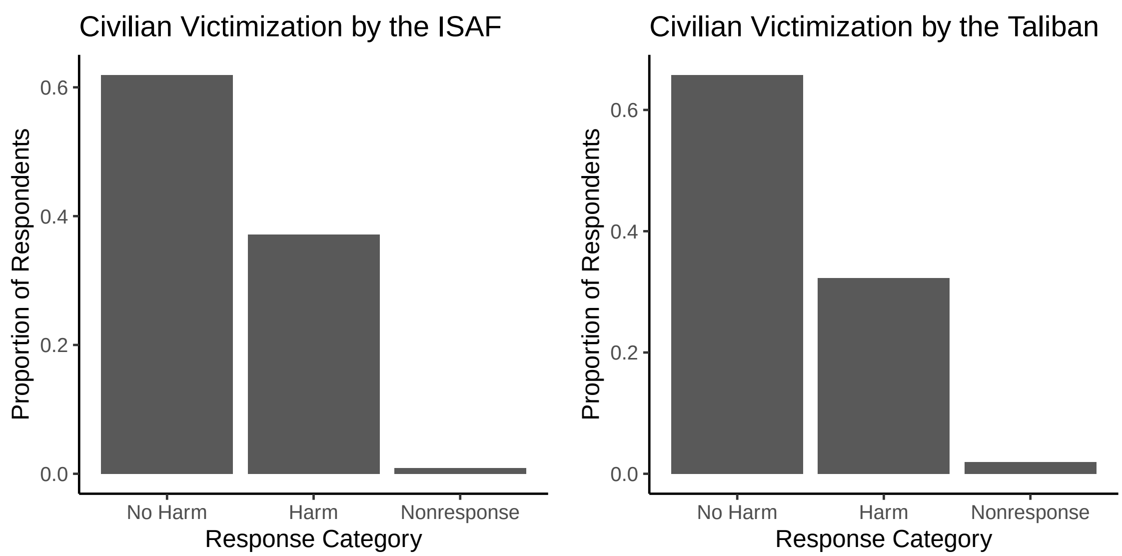
```r
## compare to the dimensions if we only delete missing for income
## instead of full listwise deletion
afghan %>%
  drop_na(income) %>%
  nrow()
```

```
## [1] 2600
```

Visualizing the Univariate Distribution

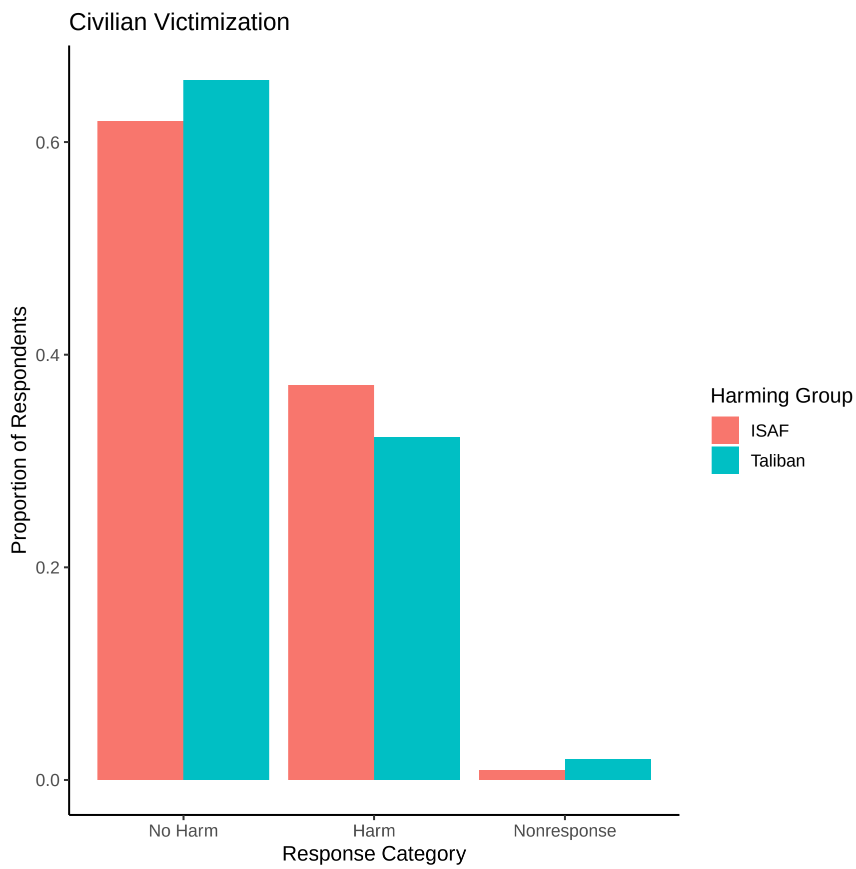
Bar plot

```
## First plot
## Bar plot with ggplot
ggplot(data = afghan, # Tell R what data to use
       aes(x = as.factor(violent.exp.ISAF))) + # specify the x-axis
  geom_bar(aes(y = ..prop.., ## add a bar plot layer
           group = 1)) +
  scale_x_discrete(labels = c('No Harm','Harm','Nonresponse')) +
  ylab("Proportion of Respondents") + # Add a label to y-axis
  xlab("Response Category") + # Add a label to the x-axis
  ggtitle("Civilian Victimization by the ISAF") # Add a title

## Second plot
## Bar plot with ggplot
ggplot(data = afghan,
       aes(x = as.factor(violent.exp.taliban))) +
  geom_bar(aes(y = ..prop..,
           group = 1)) +
  scale_x_discrete(labels = c('No Harm','Harm','Nonresponse')) +
  ylab("Proportion of Respondents") + # Add a label to y-axis
  xlab("Response Category") + # Add a label to the x-axis
  ggtitle("Civilian Victimization by the Taliban") # Add a title
```



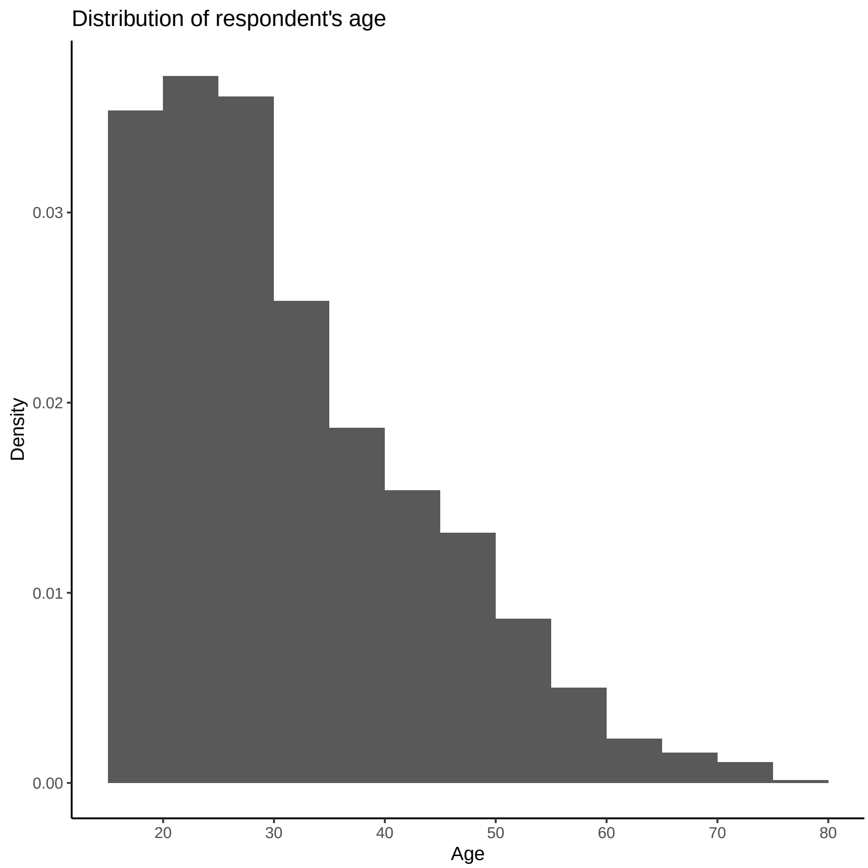

```
## reshape data longer
afghan_reshape <- afghan %>%
  pivot_longer(violent.exp.ISAF:violent.exp.taliban,
               names_to = "harming_group",
               values_to = "harm")

## Bar plot with both harm indicators together
ggplot(data = afghan_reshape, # what's different here?
       aes(x = as.factor(harm))) +
  geom_bar(aes(y = ..prop.., # what's different here?
           fill = harming_group,
           group = harming_group),
```

```
          position = "dodge") +
  scale_x_discrete(labels = c('No Harm','Harm','Nonresponse')) +
  scale_fill_discrete(name = "Harming Group", labels = c("ISAF", "Taliban")) +
  ylab("Proportion of Respondents") +
  xlab("Response Category") +
  ggtitle("Civilian Victimization")
```

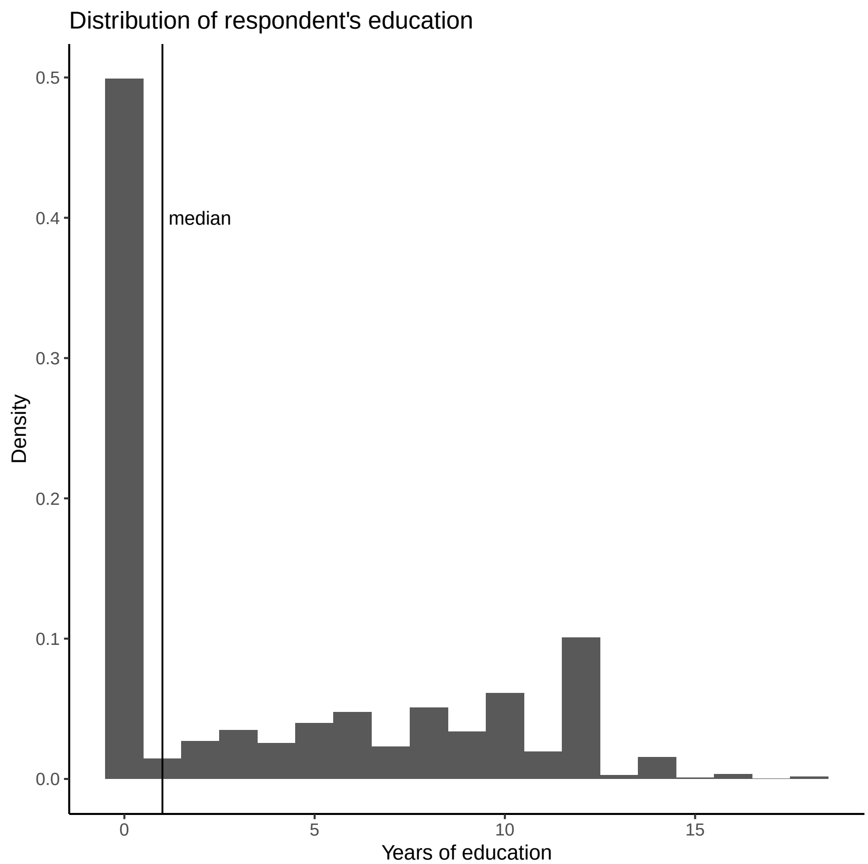


Histogram

```
ggplot(afghan, aes(x = age)) + # the data and initial aes()
  geom_histogram(aes(y = ..density..), # histogram, additional aes()
                 binwidth = 5, # how wide for each bin
                 boundary = 0) + # bin position
  scale_x_continuous(breaks = seq(20, 80, by = 10)) +
  labs(title = "Distribution of respondent's age",
       y = "Density", x = "Age") +
  theme_classic()
```



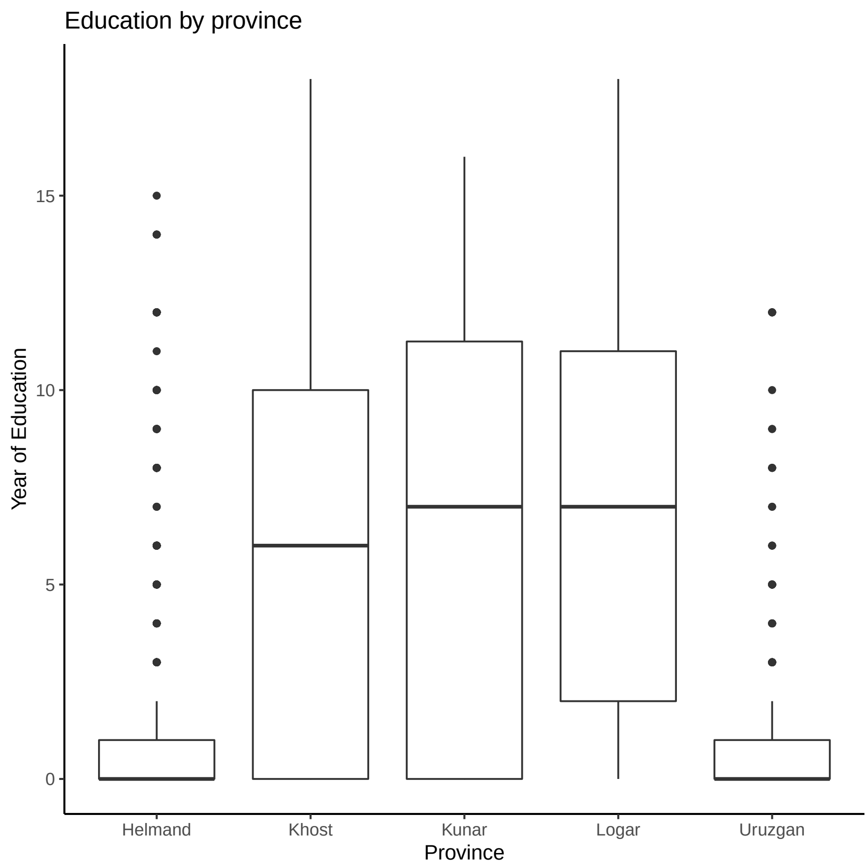

```r
ggplot(afghan, aes(x = educ.years, y = ..density..)) +
  geom_histogram(binwidth = 1, center = 0) +
  geom_vline(xintercept = median(afghan$educ.years)) +
  annotate(geom = "text", x = median(afghan$educ.years),
           y = 0.4,
           label = "median",
           hjust = -0.1) +
  labs(title = "Distribution of respondent's education",
       x = "Years of education",
       y = "Density")
```



Box plot

```
## The code for adding curly braces and text is omitted
ggplot(afghan, aes(y = age)) +
  geom_boxplot() +
  labs(y = "Age", x = "", title = "Distribution of Age")
```

```
ggplot(afghan, aes(y = educ.years, x = province)) +
  geom_boxplot() +
  labs(y = "Year of Education", x = "Province", title = "Education by province")
```




```
afghan %>%
  group_by(province) %>%
  summarize(violent.exp.taliban =
              mean(violent.exp.taliban, na.rm = TRUE),
            violent.exp.ISAF =
              mean(violent.exp.ISAF, na.rm = TRUE))
```

```
## # A tibble: 5 x 3
##   province violent.exp.taliban violent.exp.ISAF
##   <chr>                  <dbl>            <dbl>
## 1 Helmand               0.504            0.541
## 2 Khost                 0.233            0.242
## 3 Kunar                 0.303            0.399
## 4 Logar                 0.0802           0.144
## 5 Uruzgan               0.455            0.496
```

Printing and Saving Graphs

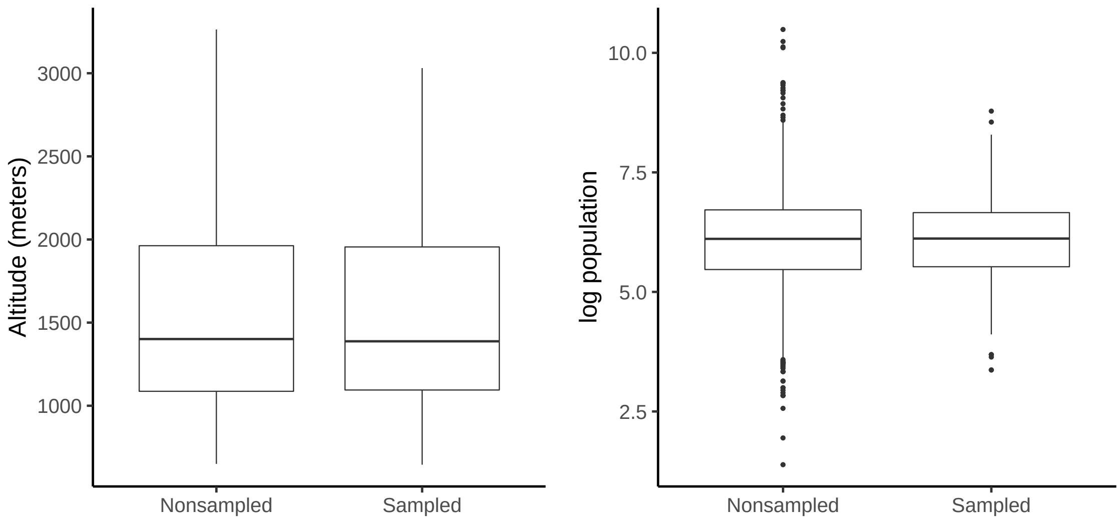
```
## Save the last figure as a pdf in the results_figures directory
ggsave("results_figures/education_by_province.pdf")
```

```r
library(gridExtra)

## The age histogram
age_hist <- ggplot(afghan, aes(x = age)) +
  geom_histogram(aes(y = ..density..),
                 binwidth = 5,
                 boundary = 0) +
  scale_x_continuous(breaks = seq(20, 80, by = 10)) +
  labs(title = "Distribution of \nrespondent's age",
       y = "Age", x = "Density")

## The education histogram
educ_hist <- ggplot(afghan, aes(x = educ.years, y = ..density..)) +
  geom_histogram(binwidth = 1, center = 0) +
  geom_vline(xintercept = median(afghan$educ.years)) +
  annotate(geom = "text", x = median(afghan$educ.years),
           y = 0.4,
           label = "median",
           hjust = -0.1) +
  labs(title = "Distribution of \nrespondent's education",
       x = "Years of education",
       y = "Density")

## Put the plots side-by-side
grid.arrange(age_hist, educ_hist, ncol = 2)
```

Survey Sampling

The Role of Randomization

```r
## Altitude box plot by sampled or not
ggplot(afghan.village, aes(x = as.factor(village.surveyed),
                           y = altitude)) +
  geom_boxplot() +
  scale_x_discrete(labels = c('Nonsampled','Sampled')) +
  labs(y = "Altitude (meters)", x = "")

## Log population box plot by sampled or not
ggplot(afghan.village, aes(x = as.factor(village.surveyed),
                           y = log(population))) +
  geom_boxplot() +
  scale_x_discrete(labels = c('Nonsampled','Sampled')) +
  labs(y = "log population", x = "")
```



Non-Response and Other Sources of Bias

```r
## Non-response rates on harm questions by province
afghan %>%
  group_by(province) %>%
  summarize(ISAF = mean(is.na(violent.exp.ISAF)),
            taliban = mean(is.na(violent.exp.taliban)))
```

```
## # A tibble: 5 x 3
##   province    ISAF taliban
##   <chr>      <dbl>   <dbl>
## 1 Helmand  0.0164  0.0304
## 2 Khost    0.00476 0.00635
## 3 Kunar    0       0
## 4 Logar    0       0
## 5 Uruzgan  0.0207  0.0620
```

```r
## Difference in mean item count between treatment/control
afghan %>%
  filter(list.group %in% c("ISAF", "control")) %>%
  group_by(list.group) %>%
  summarize(avg_list_response = mean(list.response)) %>%
  pivot_wider(names_from = list.group,
        values_from = avg_list_response) %>%
  mutate(list_response_diff = ISAF - control)
```

```
## # A tibble: 1 x 3
##   control  ISAF list_response_diff
##     <dbl> <dbl>              <dbl>
## 1    1.52  1.57             0.0490
```

```r
afghan %>%
  group_by(list.response, list.group) %>%
  count() %>%
  pivot_wider(names_from = list.group,
              values_from = n)
```

```
## # A tibble: 5 x 4
## # Groups:   list.response [5]
##   list.response control  ISAF taliban
##           <int>   <int> <int>   <int>
## 1             0     188   174      NA
## 2             1     265   278     433
## 3             2     265   260     287
## 4             3     200   182     198
## 5             4      NA    24      NA
```

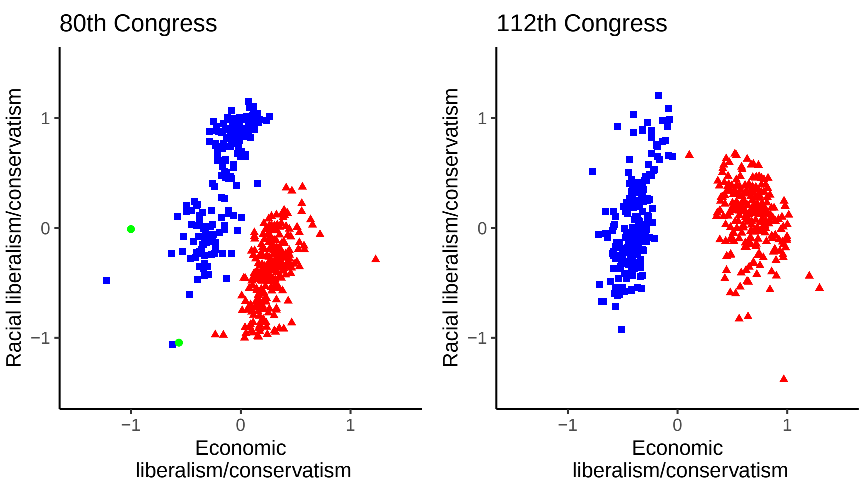
Measuring Political Polarization

Summarizing Bivariate Relationships

Scatter plot

```r
## Necessary packages and data
library(gridExtra)
data("congress", package = "qss")

## 80th congress
plot_80 <- ggplot(data = filter(congress, congress == 80),
       aes(x = dwnom1, y = dwnom2)) +
  geom_point(aes(shape = party, color = party),
             show.legend = FALSE) +
  scale_color_manual(values = c(Democrat = "blue",
                                Republican = "red",
                                Other = "green")) +
  scale_shape_manual(values = c(Democrat = "square",
                                Republican = "triangle",
                                Other = "circle")) +
  scale_y_continuous("Racial liberalism/conservatism",
                     limits = c(-1.5, 1.5)) +
  scale_x_continuous("Economic\n liberalism/conservatism",
                     limits = c(-1.5, 1.5)) +
  ggtitle("80th Congress") +
  coord_fixed()

## 112th congress
plot_112 <- ggplot(data = filter(congress, congress == 112),
       aes(x = dwnom1, y = dwnom2)) +
  geom_point(aes(shape = party, color = party),
             show.legend = FALSE) +
  scale_color_manual(values = c(Democrat = "blue",
                                Republican = "red",
```

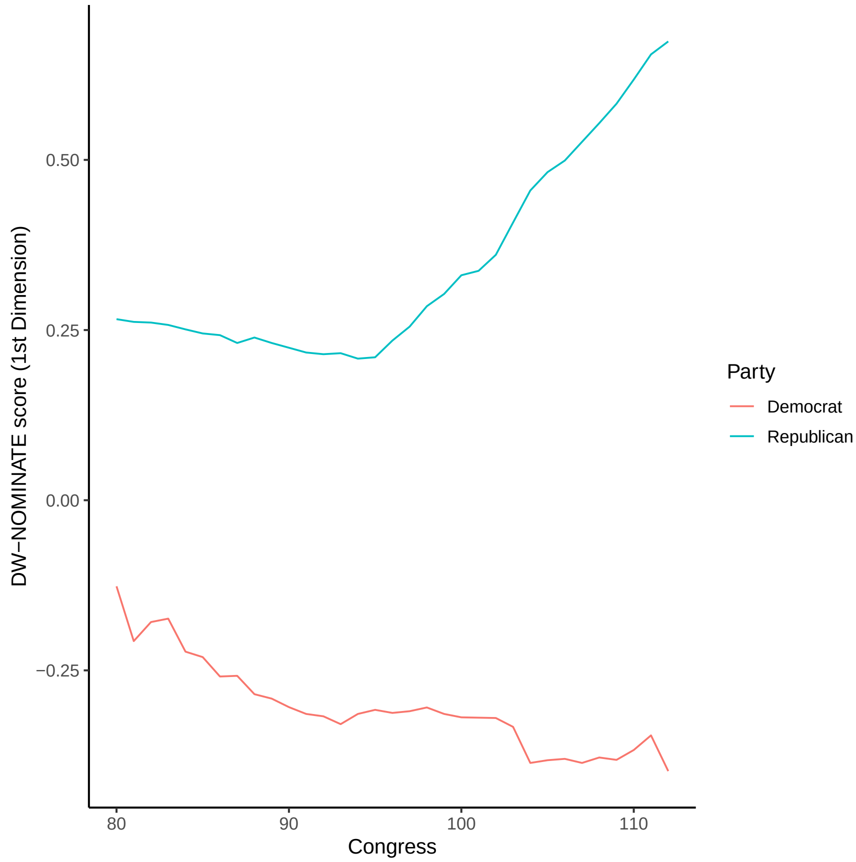
```r
                                        Other = "green")) +
    scale_shape_manual(values = c(Democrat = "square",
                                        Republican = "triangle",
                                        Other = "circle")) +
  scale_y_continuous("Racial liberalism/conservatism",
                     limits = c(-1.5, 1.5)) +
  scale_x_continuous("Economic\n liberalism/conservatism",
                     limits = c(-1.5, 1.5)) +
  ggtitle("112th Congress") +
  coord_fixed()

## Put the plots side-by-side
grid.arrange(plot_80, plot_112, ncol = 2)
```




```r
## median DWnom1 scores
median_dw1 <- congress %>%
  filter(party %in% c("Republican", "Democrat")) %>%
  group_by(party, congress) %>%
  summarize(median_dw1 = median(dwnom1))
```

```r
ggplot(data = median_dw1,
       aes(x = congress, y = median_dw1,
           color = party)) +
  geom_line() +
```

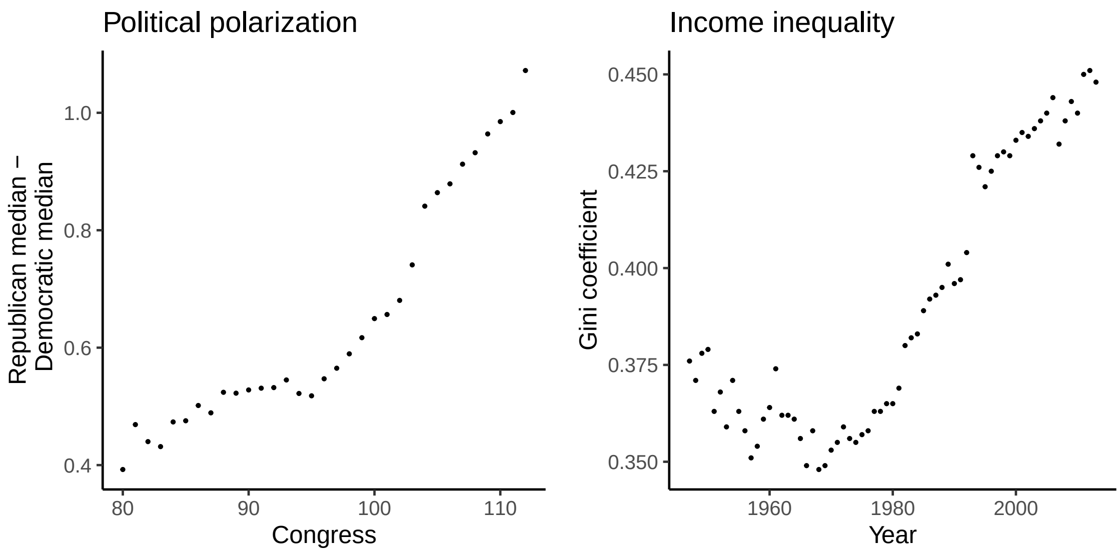
```
  labs(y = "DW-NOMINATE score (1st Dimension)", x = "Congress",
       color = "Party")
```



Correlation

```
## First, reshape the median data and calculate partisan difference
polarization <- median_dw1 %>%
  pivot_wider(names_from = party,
              values_from = median_dw1) %>%
  mutate(polarization = Republican - Democrat)

## Plot polarization over time (by congress)
ggplot(polarization, aes(x = congress, y = polarization)) +
  geom_point() +
  labs(x = "Congress", y = "Republican median -\n Democratic median") +
  ggtitle("Political polarization")

## Read in the Gini data
data("USGini", package = "qss")

## Plot US Gini over time (by year)
ggplot(USGini, aes(x = year, y = gini)) +
  geom_point() +
```

```
  labs(x = "Year", y = "Gini coefficient") +
  ggtitle("Income inequality")
```

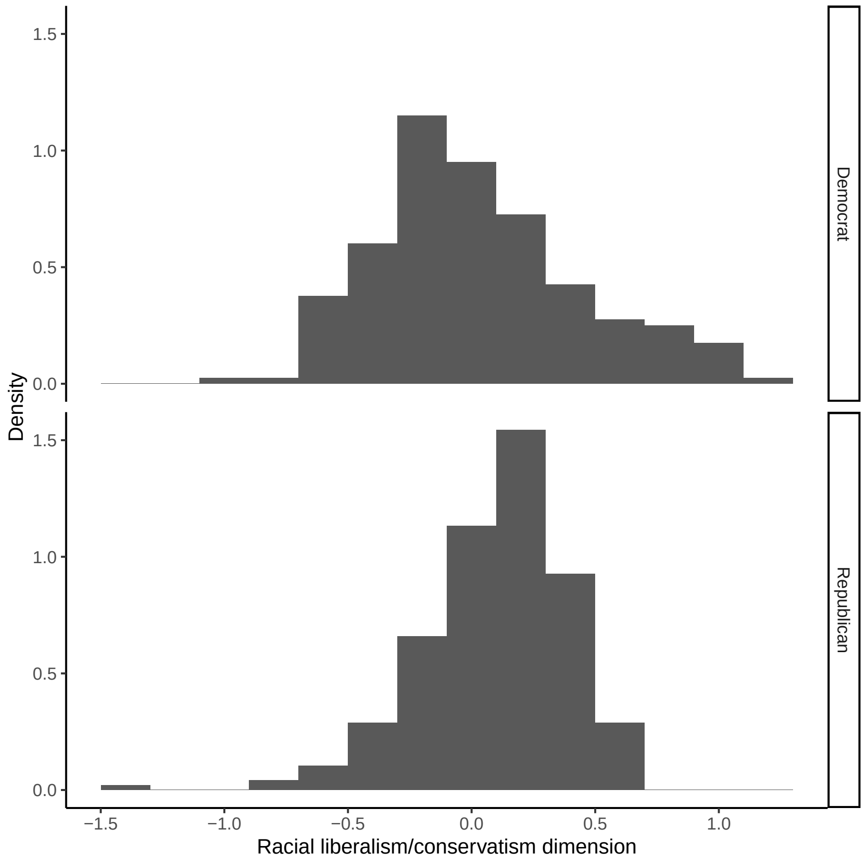



```
## Every second year Gini
gini_2yr <- USGini %>%
  filter(row_number() %% 2 == 0) %>%
  select(gini) %>%
  pull()

## Pull out the polarization score
pol_annual <- polarization %>%
  select(polarization) %>%
  pull()

## The correlation
cor(gini_2yr, pol_annual)
```

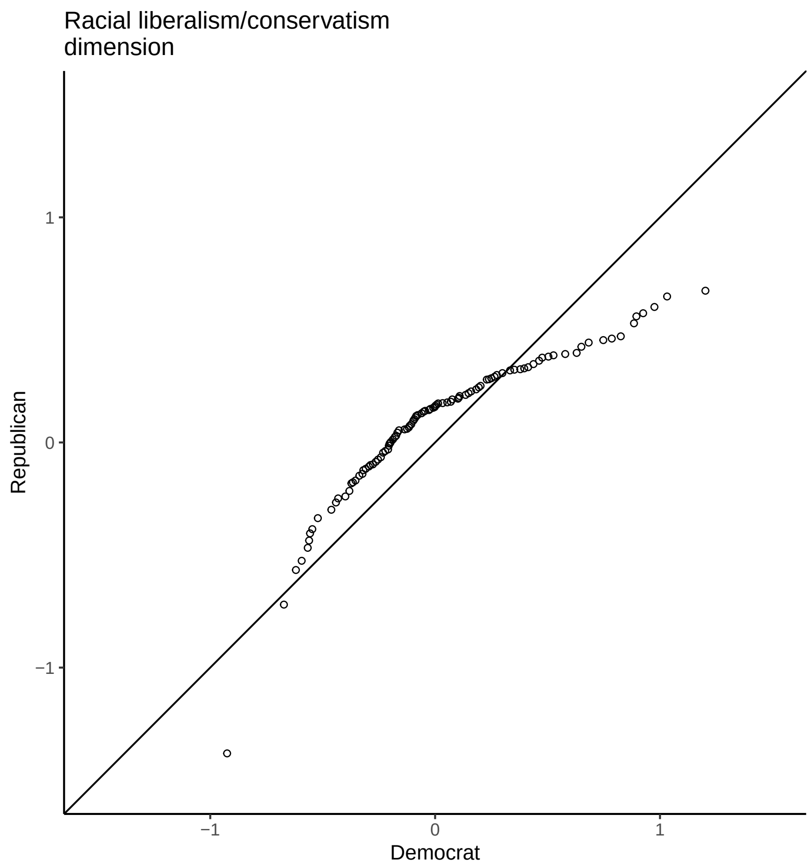
```
## [1] 0.9418128
```

Quantile-Quantile Plot

```
congress %>%
  filter(congress == 112, party %in% c("Republican", "Democrat")) %>%
  ggplot(aes(x = dwnom2, y = ..density..)) +
  geom_histogram(binwidth = .2) +
  facet_grid(party ~ .) +
  labs(x = "Racial liberalism/conservatism dimension",
       y = "Density")
```




```r
quantile_probs <- seq(from = 0, to = 1, by = 0.01)
quantile_names <- as.character(quantile_probs)

## The quantile data
quantiles <- congress %>%
  filter(congress == 112) %>%
  group_by(party) %>%
  summarize(dwnom_quantile = quantile(dwnom2, probs = quantile_probs),
            quantile = quantile_names) %>%
  pivot_wider(names_from = party,
              values_from = dwnom_quantile)
## plot it
ggplot(data = quantiles,
       aes(x = Democrat,
           y = Republican)) +
  geom_point(shape = 1) +
  ylim(-1.5, 1.5) +
  xlim(-1.5, 1.5) +
  geom_abline(intercept = 0, slope = 1) +
  ggtitle("Racial liberalism/conservatism \ndimension") +
  coord_fixed()
```




```r
## x-axis
dem112 <- filter(congress, party == "Democrat", congress == 112)
## y-axis
rep112 <- filter(congress, party == "Republican", congress == 112)

## Q-Q plot
qqplot(x = dem112$dwnom2,
       y = rep112$dwnom2,
       xlab = "Democrats",
       ylab = "Republicans",
       xlim = c(-1.5, 1.5), ylim = c(-1.5, 1.5),
       main = "Racial liberalism/conservatism dimension")
```

Clustering

Matrix in R

```r
## 3x4 matrix filled by row; first argument take actual entries
x <- matrix(1:12, nrow = 3, ncol = 4, byrow = TRUE)
rownames(x) <- c("a", "b", "c")
colnames(x) <- c("d", "e", "f", "g")
dim(x) # dimension
```

```
## [1] 3 4
```

```
x
```

```
##    d  e  f  g
## a 1  2  3  4
## b 5  6  7  8
## c 9 10 11 12
```

```
## data frame can take different data types
y <- data.frame(y1 = as.factor(c("a", "b", "c")), y2 = c(0.1, 0.2, 0.3))
class(y$y1)
```

```
## [1] "factor"
```

```
class(y$y2)
```

```
## [1] "numeric"
```

```
## as.matrix() converts both variables to character
z <- as.matrix(y)
z
```

```
##      y1  y2
## [1,] "a" "0.1"
## [2,] "b" "0.2"
## [3,] "c" "0.3"
```

```
## column sums
colSums(x)
```

```
##  d  e  f  g
## 15 18 21 24
```

```
## row means
rowMeans(x)
```

```
##    a    b    c
##  2.5  6.5 10.5
```

List in R

```
## create a list
x <- list(y1 = 1:10, y2 = c("hi", "hello", "hey"),
          y3 = data.frame(z1 = 1:3, z2 = c("good", "bad", "ugly")))
## 3 ways of extracting elements from a list
x$y1 # first element
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```r
x[[2]] # second element
```

```
## [1] "hi"    "hello" "hey"
```

```r
x[["y3"]] # third element
```

```
##   z1   z2
## 1  1 good
## 2  2  bad
## 3  3 ugly
```

```r
names(x) # names of all elements
```

```
## [1] "y1" "y2" "y3"
```

```r
length(x) # number of elements
```

```
## [1] 3
```

The k-means Algorithm

```r
## 80th congress, k = 2
k80two.out <- congress %>%
  filter(congress == 80) %>%
  select(dwnom1, dwnom2) %>%
  kmeans(centers = 2, nstart = 5)

## 112th congress, k = 2
k112two.out <- congress %>%
  filter(congress == 112) %>%
  select(dwnom1, dwnom2) %>%
  kmeans(centers = 2, nstart = 5)
```

```r
## elements of a k-means list object
names(k80two.out)
```

```
## [1] "cluster"      "centers"      "totss"
## [4] "withinss"     "tot.withinss" "betweenss"
## [7] "size"         "iter"         "ifault"
```

```r
## final centroids
k80two.out$centers
```

```
##        dwnom1     dwnom2
## 1 -0.04843704  0.7827259
## 2  0.14681029 -0.3389293
```

```r
k112two.out$centers
```

```
##       dwnom1     dwnom2
## 1  0.6776736 0.09061157
## 2 -0.3912687 0.03260696
```

```r
# load needed library
library(tidymodels) # or library(broom)

## tidy() output
k80two.clusters <- tidy(k80two.out)
k80two.clusters
```

```
## # A tibble: 2 x 5
##    dwnom1 dwnom2  size withinss cluster
##     <dbl>  <dbl> <int>    <dbl> <fct>
## 1 -0.0484  0.783   135     10.9 1
## 2  0.147  -0.339   311     54.9 2
```

```r
k112two.clusters <- tidy(k112two.out)
k112two.clusters
```

```
## # A tibble: 2 x 5
##   dwnom1 dwnom2  size withinss cluster
##    <dbl>  <dbl> <int>    <dbl> <fct>
## 1  0.678 0.0906   242     27.1 1
## 2 -0.391 0.0326   201     38.8 2
```

```r
## Members per cluster, 80th
congress80 <-
  congress %>%
  filter(congress == 80) %>%
  mutate(cluster2 = k80two.out$cluster) %>%
  group_by(party, cluster2) %>%
  count() %>%
  pivot_wider(names_from = cluster2,
              values_from = n)

## Members per cluster, 112th
congress112 <-
  congress %>%
  filter(congress == 112) %>%
  mutate(cluster2 = k112two.out$cluster) %>%
  group_by(party, cluster2) %>%
  count() %>%
  pivot_wider(names_from = cluster2,
              values_from = n)
```

```r
## 80th congress, k = 4
k80four.out <- congress %>%
  filter(congress == 80) %>%
```

```r
  select(dwnom1, dwnom2) %>%
  kmeans(centers = 4, nstart = 5)

## 112th congress, k = 4
k112four.out <- congress %>%
  filter(congress == 112) %>%
  select(dwnom1, dwnom2) %>%
  kmeans(centers = 4, nstart = 5)
```

```r
## plot the 80th congress
## prepare the data
congress80 <- filter(congress, congress == 80) %>%
  mutate(cluster4 = factor(k80four.out$cluster))

## prepare the centroids
k80four.clusters <- tidy(k80four.out)

## Plot it
ggplot() +
  geom_point(data = congress80,
             aes(x = dwnom1,
                 y = dwnom2,
                 color = cluster4)) +
  geom_point(data = k80four.clusters,
             mapping = aes(x = dwnom1, y = dwnom2),
             size = 3,
             shape = 8) +
  ylim(-1.5, 1.5) +
  xlim(-1.5, 1.5) +
  coord_fixed() +
  theme(legend.position = "none")

## plot the 112th congress
## prepare the data
congress112 <- filter(congress, congress == 112) %>%
  mutate(cluster4 = factor(k112four.out$cluster))

## prepare the centroids
k112four.clusters <- tidy(k112four.out)

## Plot it
ggplot() +
  geom_point(data = congress112,
             aes(x = dwnom1,
                 y = dwnom2,
                 color = cluster4)) +
  geom_point(data = k112four.clusters,
             mapping = aes(x = dwnom1, y = dwnom2),
             size = 3,
             shape = 8) +
  ylim(-1.5, 1.5) +
  xlim(-1.5, 1.5) +
  coord_fixed() +
```

```
theme(legend.position = "none")
```