Project2

waheeb Algabri

```
knitr:: include_graphics("player.png")
```

Player	Points	Rebounds	Assists	Steals	Blocks
LeBron James	27.4	8.5	8.3	1.3	0.6
Kevin Durant	29.0	7.3	5.3	0.7	1.2
James Harden	25.2	6.5	11.5	1.5	0.8
Giannis Antetokounmpo	28.4	11.2	5.9	1.3	1.4

untidy data posted by mohammed ramadan in discussion 5

One example of an untidy dataset is a table of NBA player statistics that is taken from an HTML page. The table is in "wide" format, where each row represents a player, and each column represents a statistic.

An analysis that might be performed on this data is to compare the performance of different players based on their statistics. For example, one might be interested in comparing the points scored by LeBron James and Kevin Durant, or in comparing the assists of James Harden and Giannis Antetokounmpo. However, the wide format of the table makes it difficult to perform these comparisons, since the relevant statistics are spread out across different columns.

```
library(RMySQL)
con <- dbConnect(MySQL(),
                 host = "localhost",
                 username = "root",
                 password = "Alex9297248844",
                 dbname = "Project2")
```

```
con <- dbGetQuery(con, "SELECT * FROM player_stats ")
```

```
print (con)
```

```
##   id                 player points rebounds assists steals blocks
## 1  1           LeBron James   27.4      8.5     8.3    1.3    0.6
## 2  2           Kevin Durant   29.0      7.3     5.3    0.7    1.2
## 3  3           James Harden   25.2      6.5    11.5    1.5    0.8
## 4  4 Giannis Antetokounmpo   28.4     11.2     5.9    1.3    1.4
```

Tidy and transform the data

```
# convert the data from wide format to long format

library(tidyr)

df <- con %>%
  pivot_longer(cols = c("points", "rebounds", "assists", "steals", "blocks"),
               names_to = "statistic",
               values_to = "value")
```

```
print(df)
```

```
## # A tibble: 20 x 4
##       id player                statistic value
##    <int> <chr>                 <chr>     <dbl>
##  1     1 LeBron James          points     27.4
##  2     1 LeBron James          rebounds    8.5
##  3     1 LeBron James          assists     8.3
##  4     1 LeBron James          steals      1.3
##  5     1 LeBron James          blocks      0.6
##  6     2 Kevin Durant          points     29
##  7     2 Kevin Durant          rebounds    7.3
##  8     2 Kevin Durant          assists     5.3
##  9     2 Kevin Durant          steals      0.7
## 10     2 Kevin Durant          blocks      1.2
## 11     3 James Harden          points     25.2
## 12     3 James Harden          rebounds    6.5
```

```
## 13     3 James Harden          assists    11.5
## 14     3 James Harden          steals      1.5
## 15     3 James Harden          blocks      0.8
## 16     4 Giannis Antetokounmpo points     28.4
## 17     4 Giannis Antetokounmpo rebounds   11.2
## 18     4 Giannis Antetokounmpo assists     5.9
## 19     4 Giannis Antetokounmpo steals      1.3
## 20     4 Giannis Antetokounmpo blocks      1.4
```

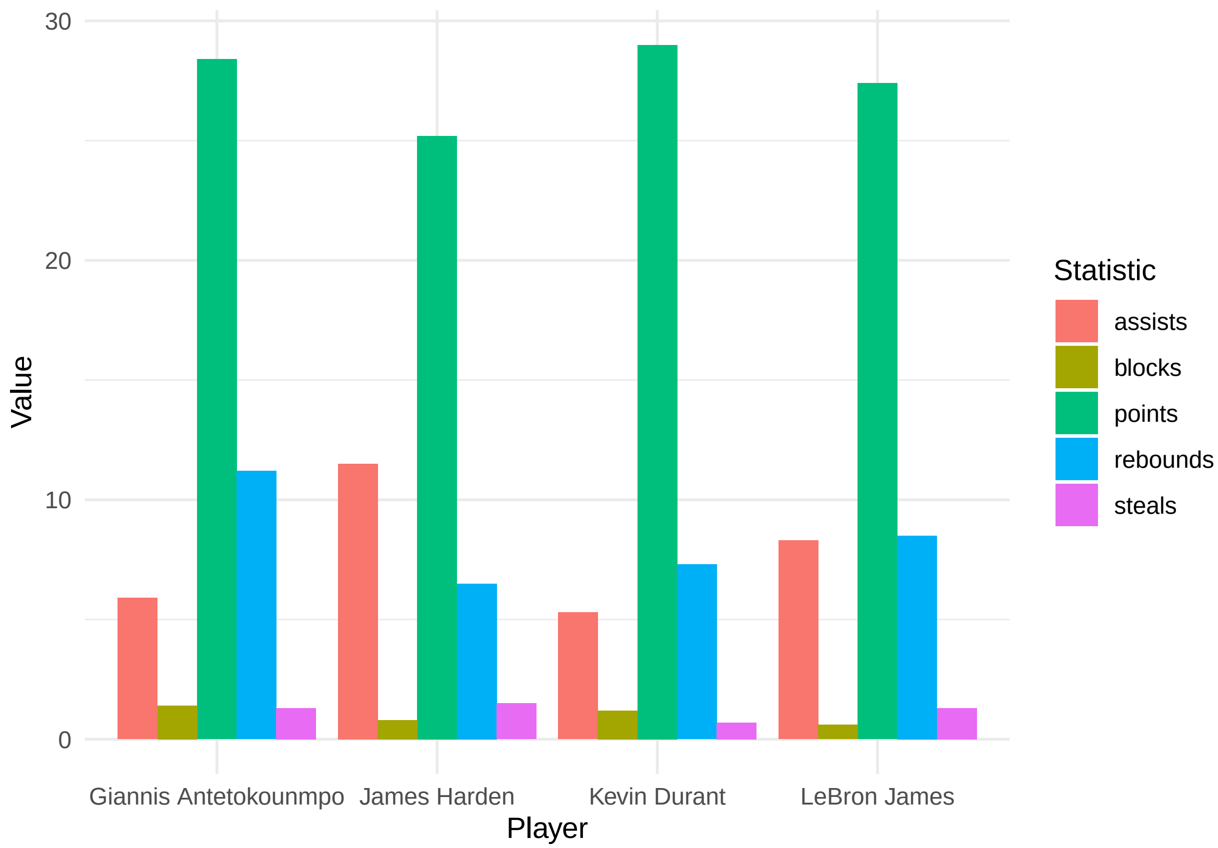
using dplyr to filter the data and compare the points scored by LeBron James and Kevin Duran

```r
library(dplyr)

df %>%
  filter(player %in% c("LeBron James", "Kevin Durant"), statistic == "points") %>%
  select(player, value)
```

```
## # A tibble: 2 x 2
##   player       value
##   <chr>        <dbl>
## 1 LeBron James  27.4
## 2 Kevin Durant  29
```

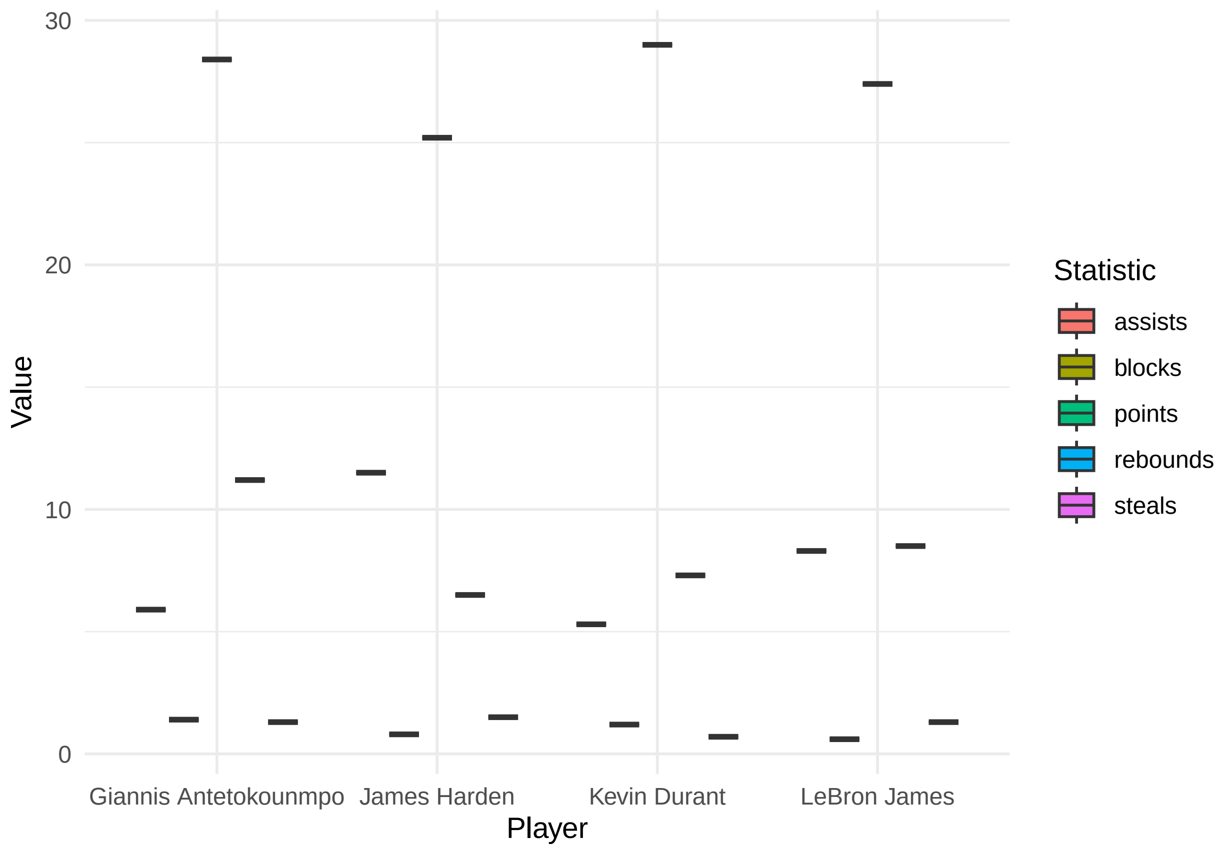
bar charts to compare the mean values of different statistics for each player

```r
library(ggplot2)

ggplot(df, aes(x = player, y = value, fill = statistic)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "Player", y = "Value", fill = "Statistic") +
  theme_minimal()
```



box plots to compare the distribution of different statistics for each player

```
ggplot(df, aes(x = player, y = value, fill = statistic)) +
  geom_boxplot() +
  labs(x = "Player", y = "Value", fill = "Statistic") +
  theme_minimal()
```




```r
# Filter the data to include only LeBron James and Kevin Durant
data_subset <- df %>%
  filter(player %in% c("LeBron James", "Kevin Durant"))

# Perform a two-sample t-test to compare the mean points scored between the two players
t.test(value ~ player, data = data_subset, var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  value by player
## t = -0.073002, df = 8, p-value = 0.9436
## alternative hypothesis: true difference in means between group Kevin Durant and group LeBron James is
## 95 percent confidence interval:
##  -16.94587  15.90587
## sample estimates:
## mean in group Kevin Durant mean in group LeBron James
##                       8.70                       9.22
```

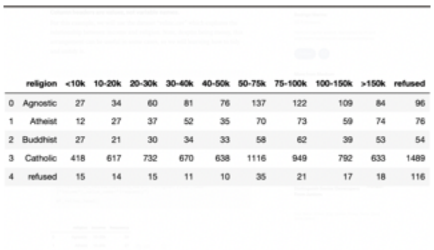
the p-value of 0.94 suggests that there is not a significant difference in the means of the two groups, and the confidence interval (-16.95, 15.91) includes zero, which also supports this conclusion.

conculution I compared the performance of different players based on their statistics, specifically focusing on the points scored by LeBron James and Kevin Durant. I also used visualizations to compare the mean values and distribution of different statistics for each player,also performed a two-sample t-test to determine whether there was a significant difference in the mean points scored between LeBron James and Kevin Durant.

Untidy data posted by Coco Donovan in discussion 5

sources: untidy data

```r
knitr:: include_graphics("coco.png")
```



	religion	<10k	10-20k	20-30k	30-40k	40-50k	50-75k	75-100k	100-150k	>150k	refused
0	Agnostic	27	34	60	81	76	137	122	109	84	96
1	Atheist	12	27	37	52	35	70	73	59	74	76
2	Buddhist	27	21	30	34	33	58	62	39	53	54
3	Catholic	418	617	732	670	638	1116	949	792	633	1489
4	refused	15	14	15	11	10	35	21	17	18	116

As far as analysis goes you could group by religion and see what the religious makeup of all respondents was by percentages. This is also just an idea, but it could be helpful to conduct a visual analysis with parallel bar charts to get and idea of which religion has the wealthiest followers (based solely on the results of this data)

```r
library(RMySQL)
# Connect to the database using the environment variables
coco<- dbConnect(MySQL(),
                 host = "localhost",
                 username = "root",
                 password = "Alex9297248844",
                 dbname = "Project2")
```

```r
coco <- dbGetQuery(coco, "SELECT * from religion_survey")
```

```r
knitr::kable(coco)
```

Data cleanup

id	religion	lt_10k	_10_20k	_20_30k	_30_40k	_40_50k	_50_75k	_75_100k	_100_150k	gt_150k	refused
1	Agnostic	27	34	60	81	76	134	122	109	84	96
2	Atheist	12	27	37	52	35	70	73	59	74	76
3	Buddhist	27	21	30	34	33	58	62	39	53	54

id	religion	lt__10k	__10__20k	__20__30k	__30__40k	__40__50k	__50__75k	__75__100k	__100__150k	gt__150k	refused
4	Catholic	418	617	732	670	638	1116	949	792	633	1489
5	Refused	15	14	15	11	10	35	21	17	18	116

```
library(tidyverse)
library(dplyr)
```

Tidy and transform the data using tidyr and dplyr drop id column

```
coco_new <- select(coco, -id)
```

```
knitr::kable(coco_new)
```

religion	lt__10k	__10__20k	__20__30k	__30__40k	__40__50k	__50__75k	__75__100k	__100__150k	gt__150k	refused
Agnostic	27	34	60	81	76	134	122	109	84	96
Atheist	12	27	37	52	35	70	73	59	74	76
Buddhist	27	21	30	34	33	58	62	39	53	54
Catholic	418	617	732	670	638	1116	949	792	633	1489
Refused	15	14	15	11	10	35	21	17	18	116

```
coco_tidy <- coco_new %>%
  pivot_longer(cols = -religion, names_to = "income_level", values_to = "count") %>%
  mutate(income_level = gsub("income_", "", income_level)) %>%
  arrange(religion, income_level)

head (coco_tidy)
```

Convert the data from wide format to long format using

```
## # A tibble: 6 x 3
##   religion income_level count
##   <chr>    <chr>        <int>
## 1 Agnostic _100_150k      109
## 2 Agnostic _10_20k         34
## 3 Agnostic _20_30k         60
## 4 Agnostic _30_40k         81
## 5 Agnostic _40_50k         76
## 6 Agnostic _50_75k        134
```

```
summary(coco_tidy)
```

```
##    religion         income_level           count
##  Length:50          Length:50          Min.   :  10.00
##  Class :character   Class :character   1st Qu.:  27.75
```

```
##  Mode  :character   Mode  :character   Median :  58.50
##                                        Mean   : 201.50
##                                        3rd Qu.: 114.25
##                                        Max.   :1489.00
```

Replace all occurrences of "refused" in the income_level column with NA, creating a tidy dataset where all values are of the same type and format.

```
coco_tidy <- coco_tidy %>%
  mutate(income_level = replace(income_level, income_level == "refused", NA))
```

```
head(coco_tidy)
```

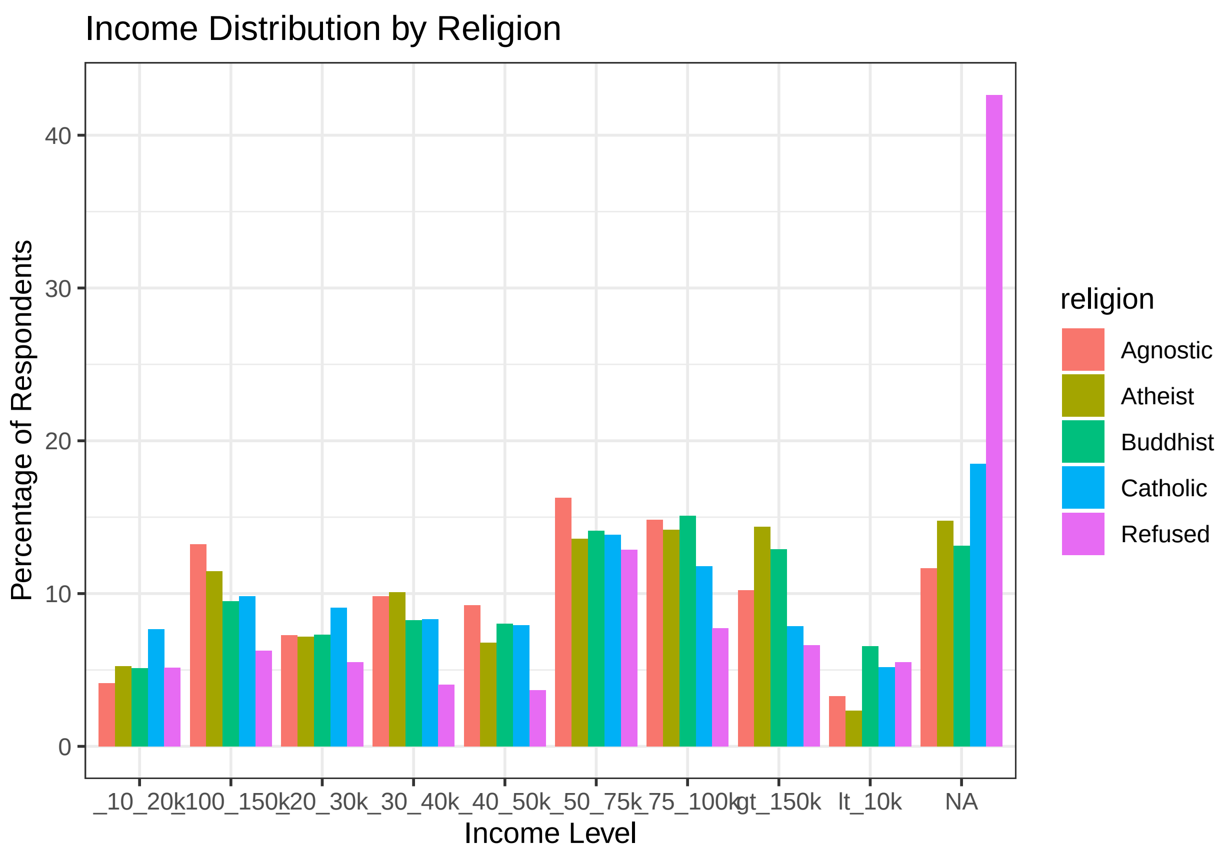
```
## # A tibble: 6 x 3
##   religion income_level count
##   <chr>    <chr>        <int>
## 1 Agnostic _100_150k      109
## 2 Agnostic _10_20k         34
## 3 Agnostic _20_30k         60
## 4 Agnostic _30_40k         81
## 5 Agnostic _40_50k         76
## 6 Agnostic _50_75k        134
```

```
library(dplyr)
library(ggplot2)

# group data by religion and income level, calculate percentage of respondents
data_summary <- coco_tidy %>%
  group_by(religion, income_level) %>%
  summarize(count = sum(count)) %>%
  mutate(percent = count/sum(count) * 100)
```

```
head(data_summary)
```

```
## # A tibble: 6 x 4
## # Groups:   religion [1]
##   religion income_level count percent
##   <chr>    <chr>        <int>   <dbl>
## 1 Agnostic _100_150k      109   13.2
## 2 Agnostic _10_20k         34    4.13
## 3 Agnostic _20_30k         60    7.29
## 4 Agnostic _30_40k         81    9.84
## 5 Agnostic _40_50k         76    9.23
## 6 Agnostic _50_75k        134   16.3
```

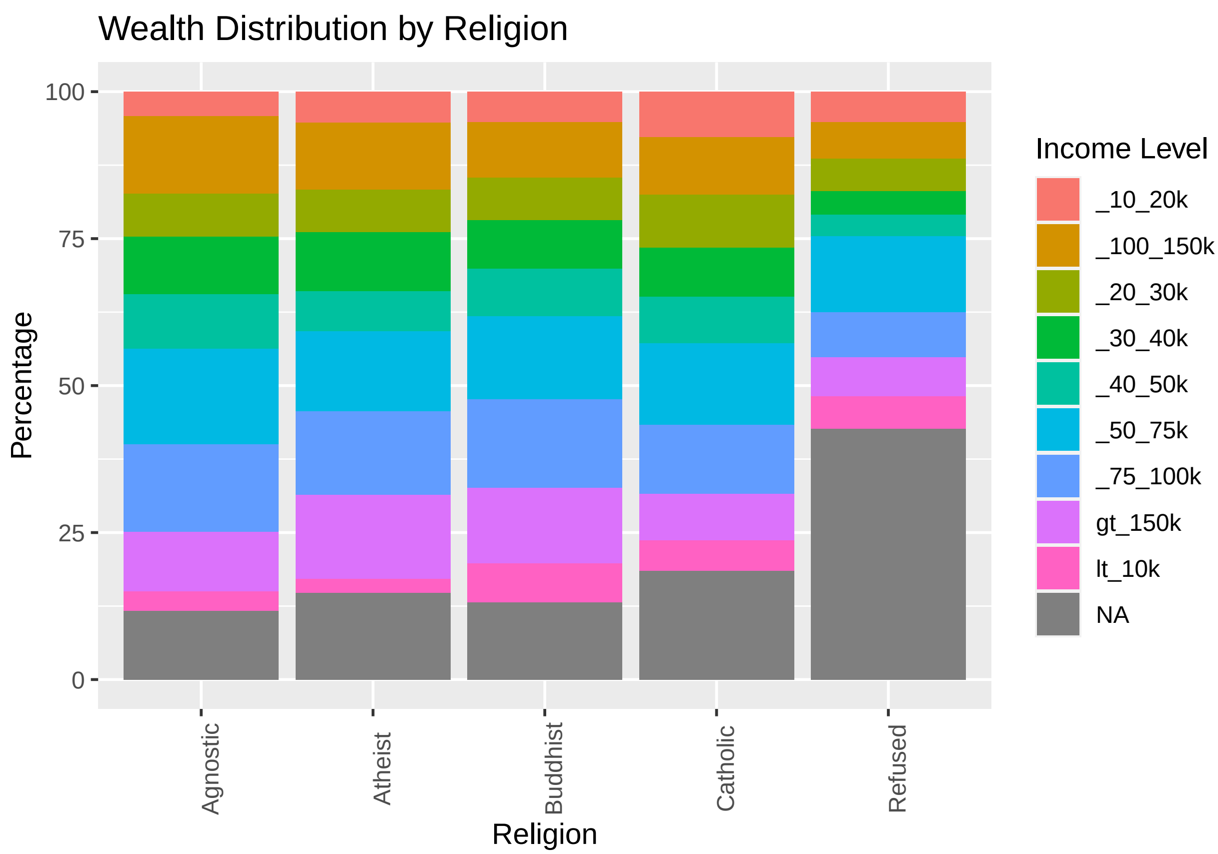
```
# create parallel bar chart
ggplot(data_summary, aes(x = income_level, y = percent, fill = religion)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Income Distribution by Religion",
       x = "Income Level", y = "Percentage of Respondents") +
  theme_bw()
```




```
summary(data_summary)
```

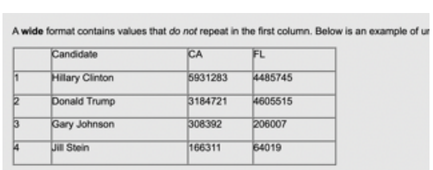
```
##    religion          income_level           count              percent
##  Length:50          Length:50          Min.   :  10.00   Min.   : 2.330
##  Class :character   Class :character   1st Qu.:  27.75   1st Qu.: 6.581
##  Mode  :character   Mode  :character   Median :  58.50   Median : 8.704
##                                        Mean   : 201.50   Mean   :10.000
##                                        3rd Qu.: 114.25   3rd Qu.:13.078
##                                        Max.   :1489.00   Max.   :42.647
```

```
# Create the stacked bar chart
ggplot(data_summary, aes(x = religion, y = percent, fill = income_level)) +
  geom_bar(stat = "identity", position = "stack") +
  theme(axis.text.x = element_text(angle = 90)) +
  labs(title = "Wealth Distribution by Religion", x = "Religion", y = "Percentage", fill = "Income Level
```



Untidy data posted by Farhana Akther in discussion 5

```
knitr:: include_graphics("Farhana.png")
```



A **wide** format contains values that *do not* repeat in the first column. Below is an example of ur

	Candidate	CA	FL
1	Hillary Clinton	5931283	4485745
2	Donald Trump	3184721	4605515
3	Gary Johnson	308392	206007
4	Jill Stein	166311	64019

The table above reports vote counts for two US states, California and Florida. In this table, the column names CA and FL are values of the variable state. Therefore, we can say that this table is in an untidy format.

As per analysis, we can compare the vote counts in each state vs candidate. We can also "melt" the table and transform into a long format and tidy the table so that further analysis can be done; including plotting functions, hypothesis testing functions, and modeling functions such as linear regression.

Source

```r
library(RMySQL)
Farhana<- dbConnect(MySQL(),
                 host = "localhost",
                 username = "root",
                 password = "Alex9297248844",
                 dbname = "Project2")
```

```r
Farhana <- dbGetQuery(Farhana, "SELECT * FROM election_results ")
```

```r
knitr::kable(Farhana)
```

Data cleanup

candidate	CA	FL
Donald Trump	3184721	4605515
Gary Johnson	308392	206007
Hillary Clinton	5931283	4485745
Jill Stein	166311	64019

Upload requied packages

```r
library(tidyverse)
library(dplyr)
```

```r
election_data <-Farhana %>%
  pivot_longer(cols = c(CA, FL), names_to = "state", values_to = "votes")
```

```r
knitr::kable(election_data)
```

Convert the data from wide to long format

candidate	state	votes
Donald Trump	CA	3184721
Donald Trump	FL	4605515
Gary Johnson	CA	308392
Gary Johnson	FL	206007
Hillary Clinton	CA	5931283
Hillary Clinton	FL	4485745
Jill Stein	CA	166311

candidate	state	votes
Jill Stein	FL	64019

Analysis Calculate the total votes for each candidate

```
total_votes <- election_data %>%
  group_by(candidate) %>%
  summarize(total_votes = sum(votes))
```

Join the total votes to the long format data frame

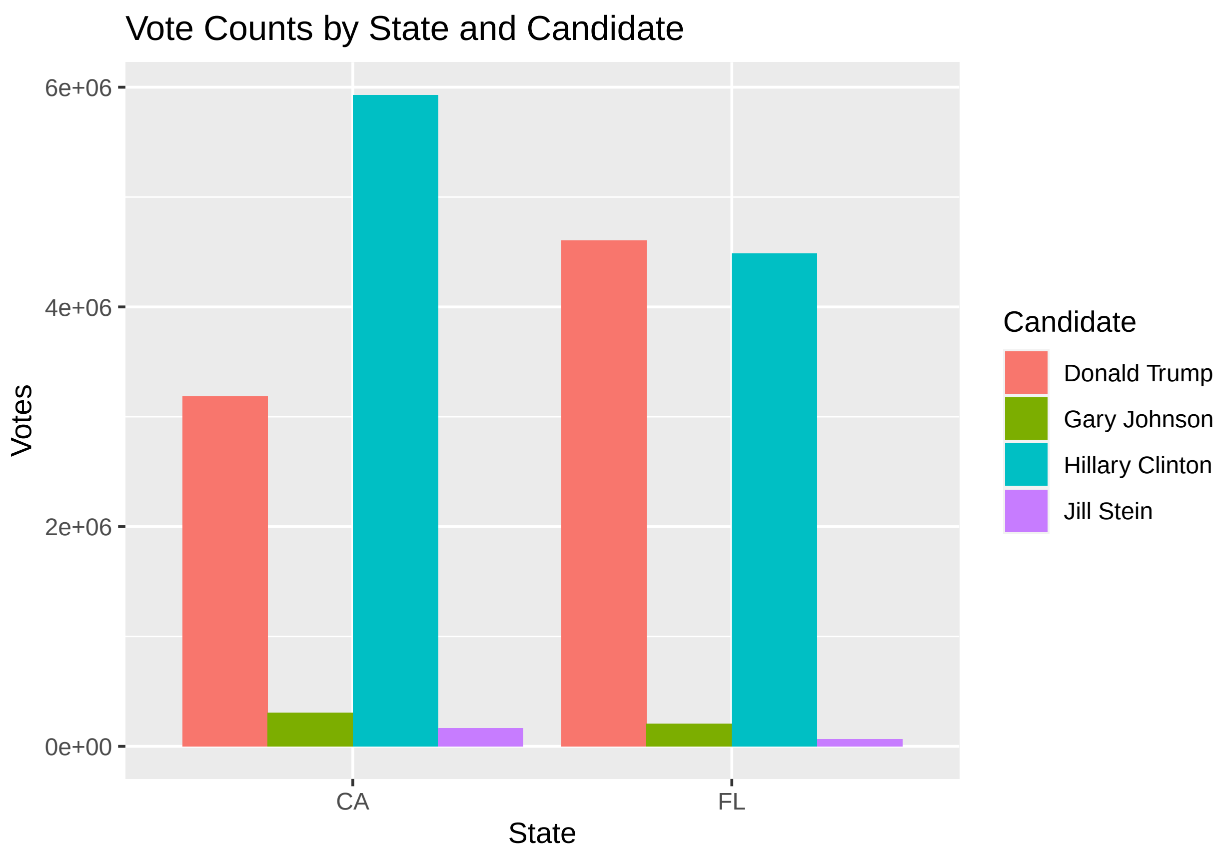
```
long_election_data <- election_data %>%
  left_join(total_votes, by = "candidate")
```

```
knitr::kable(long_election_data)
```

candidate	state	votes	total_votes
Donald Trump	CA	3184721	7790236
Donald Trump	FL	4605515	7790236
Gary Johnson	CA	308392	514399
Gary Johnson	FL	206007	514399
Hillary Clinton	CA	5931283	10417028
Hillary Clinton	FL	4485745	10417028
Jill Stein	CA	166311	230330
Jill Stein	FL	64019	230330

Bar plot of the vote counts for each candidate in each state

```
library(ggplot2)

ggplot(long_election_data, aes(x = state, y = votes, fill = candidate)) +
  geom_col(position = "dodge") +
  labs(title = "Vote Counts by State and Candidate",
       x = "State", y = "Votes",
       fill = "Candidate")
```




```
summary(long_election_data)
```

```
##   candidate             state                 votes           total_votes
##  Length:8           Length:8           Min.   :  64019   Min.   :  230330
##  Class :character   Class :character   1st Qu.: 196083   1st Qu.:  443382
##  Mode  :character   Mode  :character   Median :1746556   Median : 4152318
##                                        Mean   :2368999   Mean   : 4737998
##                                        3rd Qu.:4515688   3rd Qu.: 8446934
##                                        Max.   :5931283   Max.   :10417028
```

I will use linear regression to model the relationship between the vote counts for each candidate in California and Florida:

```
# Fit a linear regression model to the data
lm_model <- lm(votes ~ state + candidate, data = long_election_data)

# View the model summary
summary(lm_model)
```

```
##
## Call:
## lm(formula = votes ~ state + candidate, data = long_election_data)
##
## Residuals:
```

```
##       1       2       3       4       5       6       7       8
## -739075  739075   22515  -22515  694091 -694091   22468  -22468
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               3923796     654792   5.992  0.00931 **
## stateFL                    -57355     585664  -0.098  0.92816
## candidateGary Johnson    -3637918     828254  -4.392  0.02187 *
## candidateHillary Clinton  1313396     828254   1.586  0.21098
## candidateJill Stein      -3779953     828254  -4.564  0.01973 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 828300 on 3 degrees of freedom
## Multiple R-squared:  0.9509, Adjusted R-squared:  0.8855
## F-statistic: 14.53 on 4 and 3 DF,  p-value: 0.02639
```

Based on the linear regression analysis, we can see that the coefficients for the candidate variables are all significant with p-values less than 0.05, except for the stateFL variable which is not significant. This suggests that the candidate chosen had a significant effect on the number of votes received, while the state did not have a significant effect. The R-squared value of 0.9509 indicates that the model explains a high proportion of the variance in the data, and the F-statistic of 14.53 with a p-value of 0.02639 suggests that the model is statistically significant. Overall, the analysis suggests that the choice of candidate had a significant impact on the number of votes received, while the state did not have a significant effect.