Project2

Josef Waples

10/2/2020

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
twentyfifteen <- read.csv('~/Desktop/2015.csv')
twentysixteen <- read.csv('~/Desktop/2016.csv')
twentyseventeen <- read.csv('~/Desktop/2017.csv')
twentyeighteen <- read.csv('~/Desktop/2018.csv')
twentynineteen <- read.csv('~/Desktop/2019.csv')

length(twentyfifteen)
```

```
## [1] 12
```

```
length(twentysixteen)
```

```
## [1] 13
```

```
length(twentyseventeen)
```

```
## [1] 12
```

```
length(twentyeighteen)
```

```
## [1] 9
```

```
length(twentynineteen)
```

```
## [1] 9
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.6.2
```

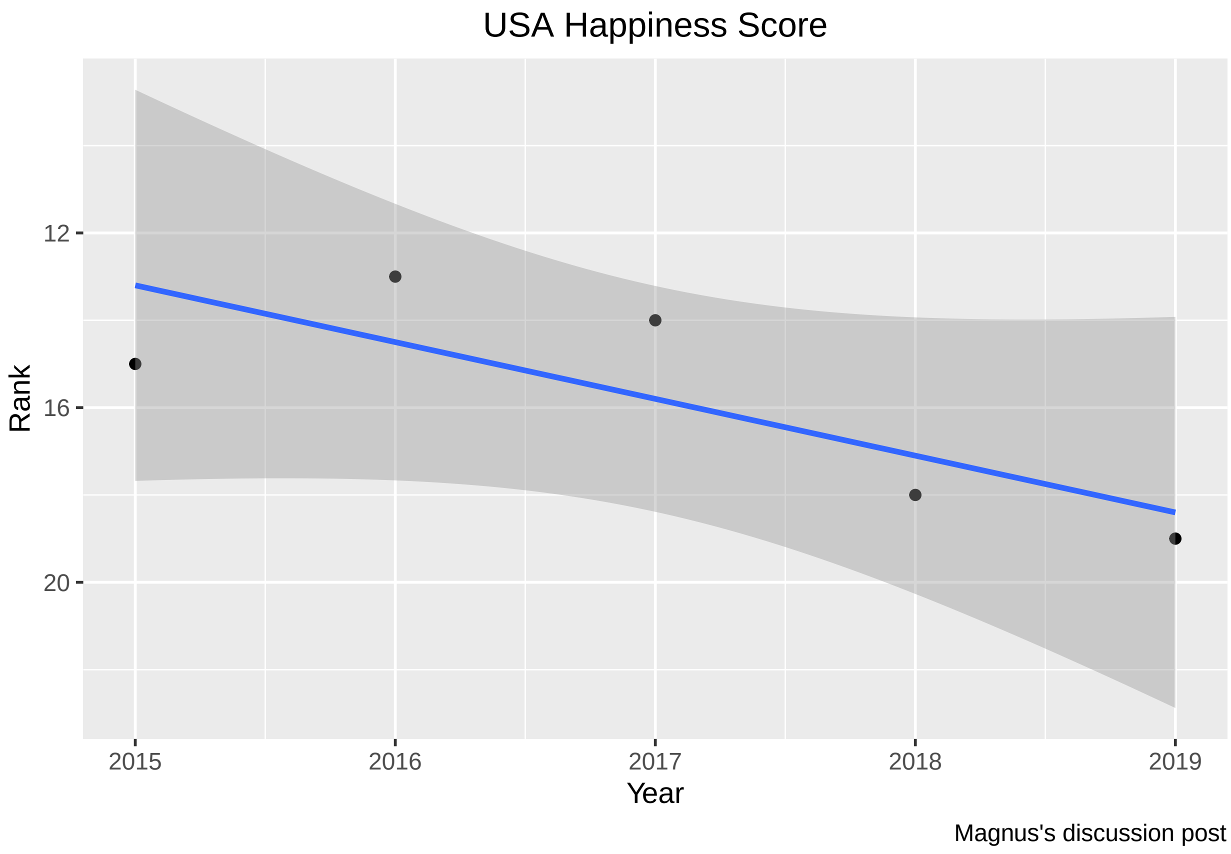
```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
# these data frames have different lengths! Unable to rbind right now. Let's just select 3 columns

twentyfifteen2 <- twentyfifteen %>%
  select(Country, Happiness.Rank, Happiness.Score)

twentysixteen2 <- twentysixteen %>%
  select(Country, Happiness.Rank, Happiness.Score)

twentyseventeen2 <- twentyseventeen %>%
  select(Country, Happiness.Rank, Happiness.Score)

twentyeighteen2 <- twentyeighteen %>%
  select(Country.or.region, Overall.rank, Score)

twentynineteen2 <- twentynineteen %>%
  select(Country.or.region, Overall.rank, Score)

# now I need to change column names in order to do an rbind

colnames(twentyfifteen2) <- c("Country", "Rank", "Score")
colnames(twentysixteen2) <- c("Country", "Rank", "Score")
colnames(twentyseventeen2) <- c("Country", "Rank", "Score")
colnames(twentyeighteen2) <- c("Country", "Rank", "Score")
colnames(twentynineteen2) <- c("Country", "Rank", "Score")

# now I need to add a new column to each in order to not lose track of the year

twentyfifteen2$Year <- 2015
twentysixteen2$Year <- 2016
twentyseventeen2$Year <- 2017
twentyeighteen2$Year <- 2018
twentynineteen2$Year <- 2019


fifteen_to_nineteen <- rbind(twentyfifteen2, twentysixteen2, twentyseventeen2, twentyeighteen2, twentyni


usa_happiness_score <- fifteen_to_nineteen %>%
  filter(Country == "United States")


canada_happiness_score <- fifteen_to_nineteen %>%
  filter(Country == "Canada")


mexico_happiness_score <- fifteen_to_nineteen %>%
  filter(Country == "Mexico")

library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.2
```

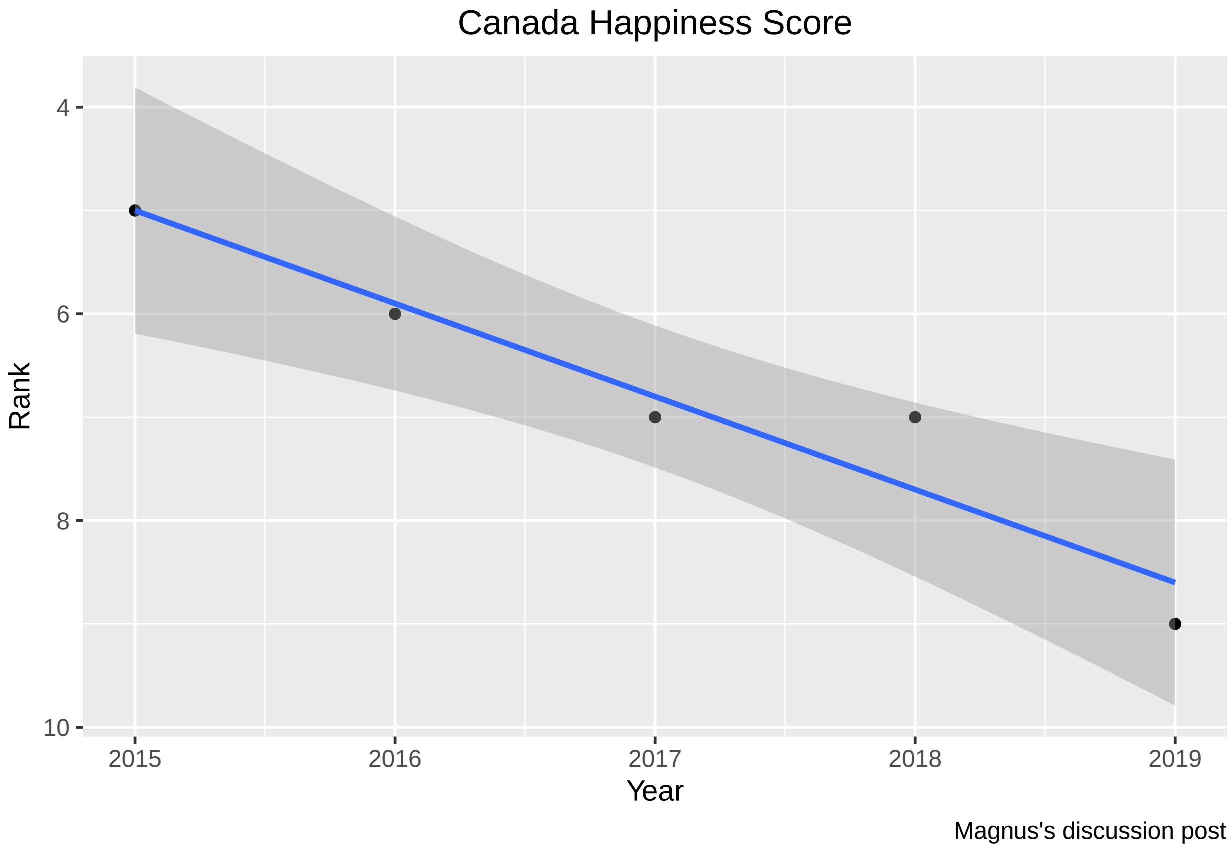
```
ggplot(usa_happiness_score, aes(x = Year, y = Rank)) +
  geom_point() +
  geom_smooth(method = "lm") +
  scale_y_continuous(trans = "reverse") +
  labs(title="USA Happiness Score") +
theme(plot.title = element_text(hjust = 0.5)) +
labs(caption = "Magnus's discussion post")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



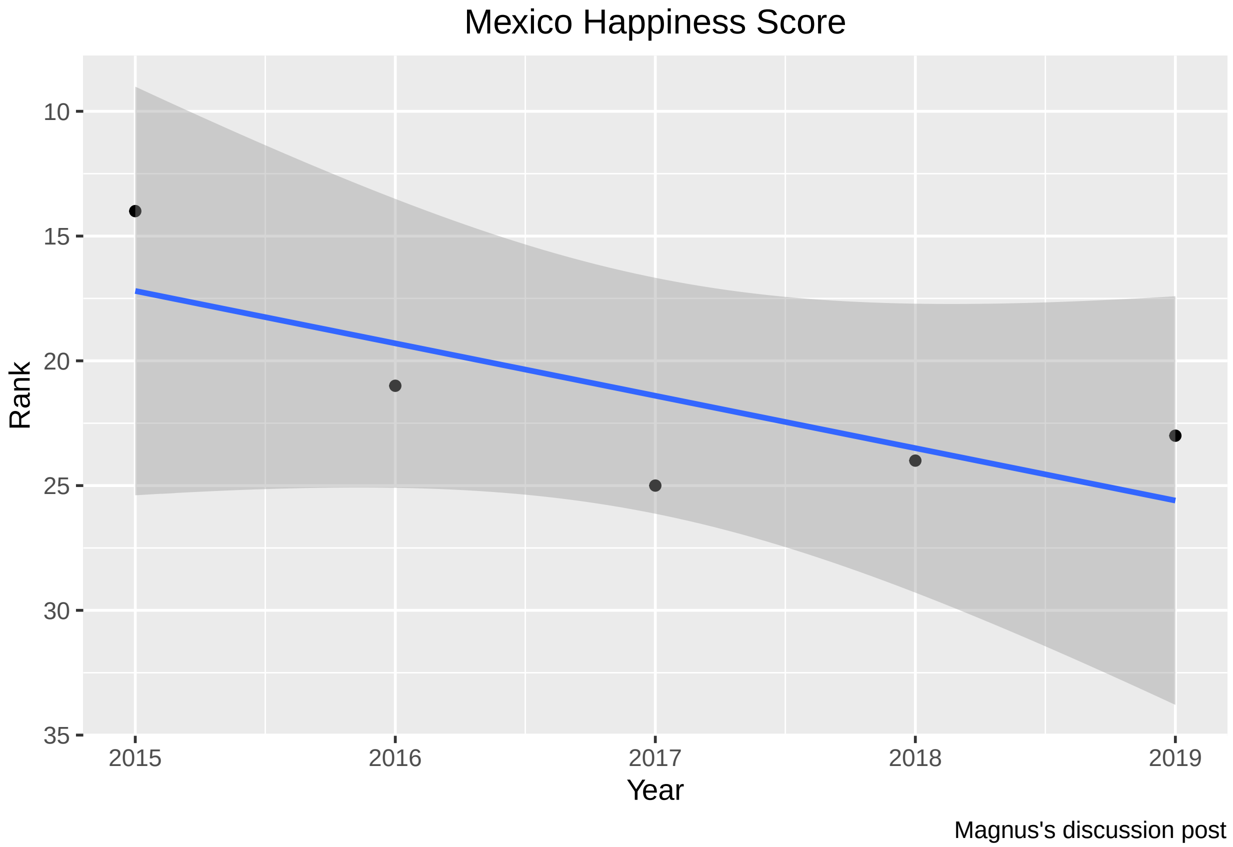
```
ggplot(canada_happiness_score, aes(x = Year, y = Rank)) +
  geom_point() +
  geom_smooth(method = "lm") +
  scale_y_continuous(trans = "reverse") +
  labs(title="Canada Happiness Score") +
theme(plot.title = element_text(hjust = 0.5)) +
labs(caption = "Magnus's discussion post")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
ggplot(mexico_happiness_score, aes(x = Year, y = Rank)) +
  geom_point() +
  geom_smooth(method = "lm") +
  scale_y_continuous(trans = "reverse") +
  labs(title="Mexico Happiness Score") +
theme(plot.title = element_text(hjust = 0.5)) +
labs(caption = "Magnus's discussion post")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

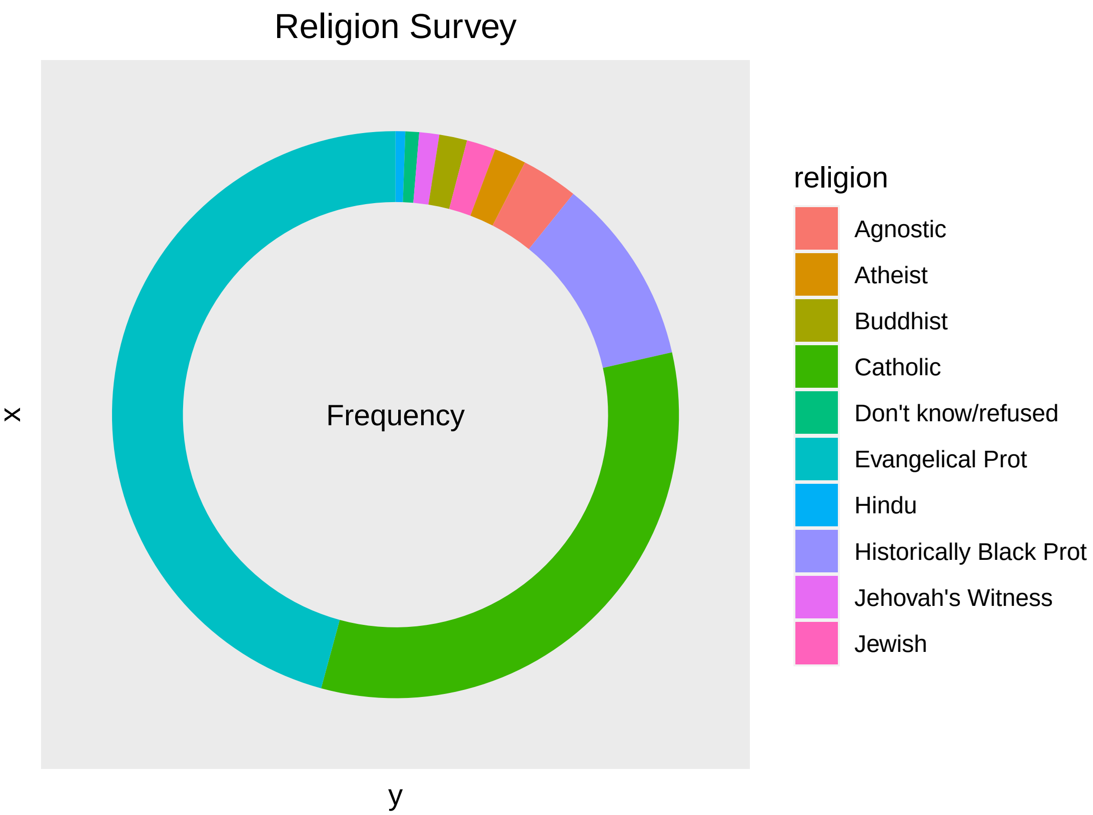


```
# Happiness Scores are getting worse in North America!
```

Including Plots

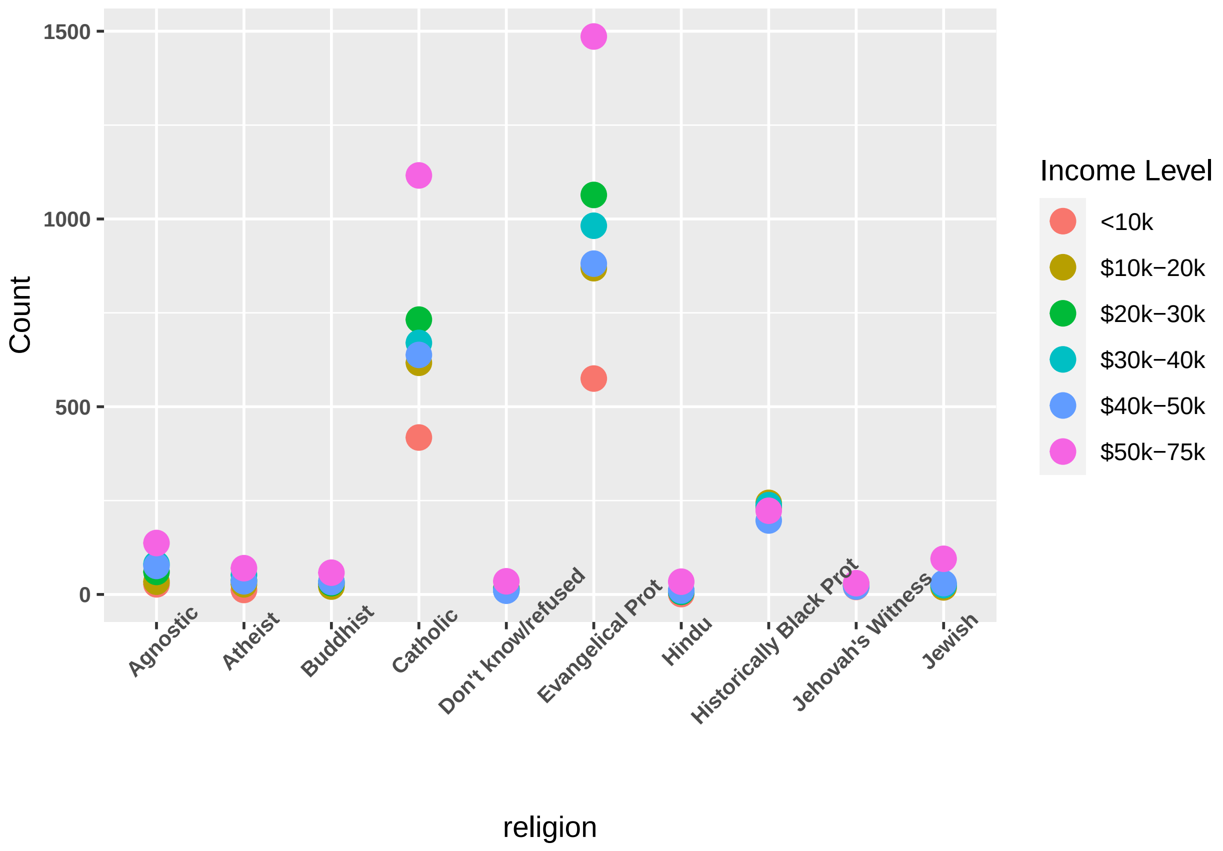
You can also embed plots, for example:

```
## Warning: ‘data_frame()‘ is deprecated as of tibble 1.1.0.
## Please use ‘tibble()‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_warnings()‘ to see where this warning was generated.
```



John's discussion post

```
## Warning: package 'tidyr' was built under R version 3.6.2

## tibble [60 x 3] (S3: tbl_df/tbl/data.frame)
##  $ religion    : chr [1:60] "Agnostic" "Atheist" "Buddhist" "Catholic" ...
##  $ Income Level: chr [1:60] "<10k" "<10k" "<10k" "<10k" ...
##  $ Count       : num [1:60] 27 12 27 418 15 575 1 228 20 19 ...
```



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   `Area Abbreviation` = col_character(),
##   Area = col_character(),
##   Item = col_character(),
##   Element = col_character(),
##   Unit = col_character()
## )

## See spec(...) for full column specifications.

## # A tibble: 6 x 63
##   `Area Abbreviat~ `Area Code` Area  `Item Code` Item  `Element Code` Element
##   <chr>                  <dbl> <chr>       <dbl> <chr>          <dbl> <chr>
## 1 AFG                        2 Afgh~        2511 Whea~           5142 Food
## 2 AFG                        2 Afgh~        2805 Rice~           5142 Food
## 3 AFG                        2 Afgh~        2513 Barl~           5521 Feed
## 4 AFG                        2 Afgh~        2513 Barl~           5142 Food
## 5 AFG                        2 Afgh~        2514 Maiz~           5521 Feed
## 6 AFG                        2 Afgh~        2514 Maiz~           5142 Food
## # ... with 56 more variables: Unit <chr>, latitude <dbl>, longitude <dbl>,
## #   Y1961 <dbl>, Y1962 <dbl>, Y1963 <dbl>, Y1964 <dbl>, Y1965 <dbl>,
## #   Y1966 <dbl>, Y1967 <dbl>, Y1968 <dbl>, Y1969 <dbl>, Y1970 <dbl>,
## #   Y1971 <dbl>, Y1972 <dbl>, Y1973 <dbl>, Y1974 <dbl>, Y1975 <dbl>,
## #   Y1976 <dbl>, Y1977 <dbl>, Y1978 <dbl>, Y1979 <dbl>, Y1980 <dbl>,
## #   Y1981 <dbl>, Y1982 <dbl>, Y1983 <dbl>, Y1984 <dbl>, Y1985 <dbl>,
## #   Y1986 <dbl>, Y1987 <dbl>, Y1988 <dbl>, Y1989 <dbl>, Y1990 <dbl>,
## #   Y1991 <dbl>, Y1992 <dbl>, Y1993 <dbl>, Y1994 <dbl>, Y1995 <dbl>,
## #   Y1996 <dbl>, Y1997 <dbl>, Y1998 <dbl>, Y1999 <dbl>, Y2000 <dbl>,
## #   Y2001 <dbl>, Y2002 <dbl>, Y2003 <dbl>, Y2004 <dbl>, Y2005 <dbl>,
## #   Y2006 <dbl>, Y2007 <dbl>, Y2008 <dbl>, Y2009 <dbl>, Y2010 <dbl>,
## #   Y2011 <dbl>, Y2012 <dbl>, Y2013 <dbl>
```

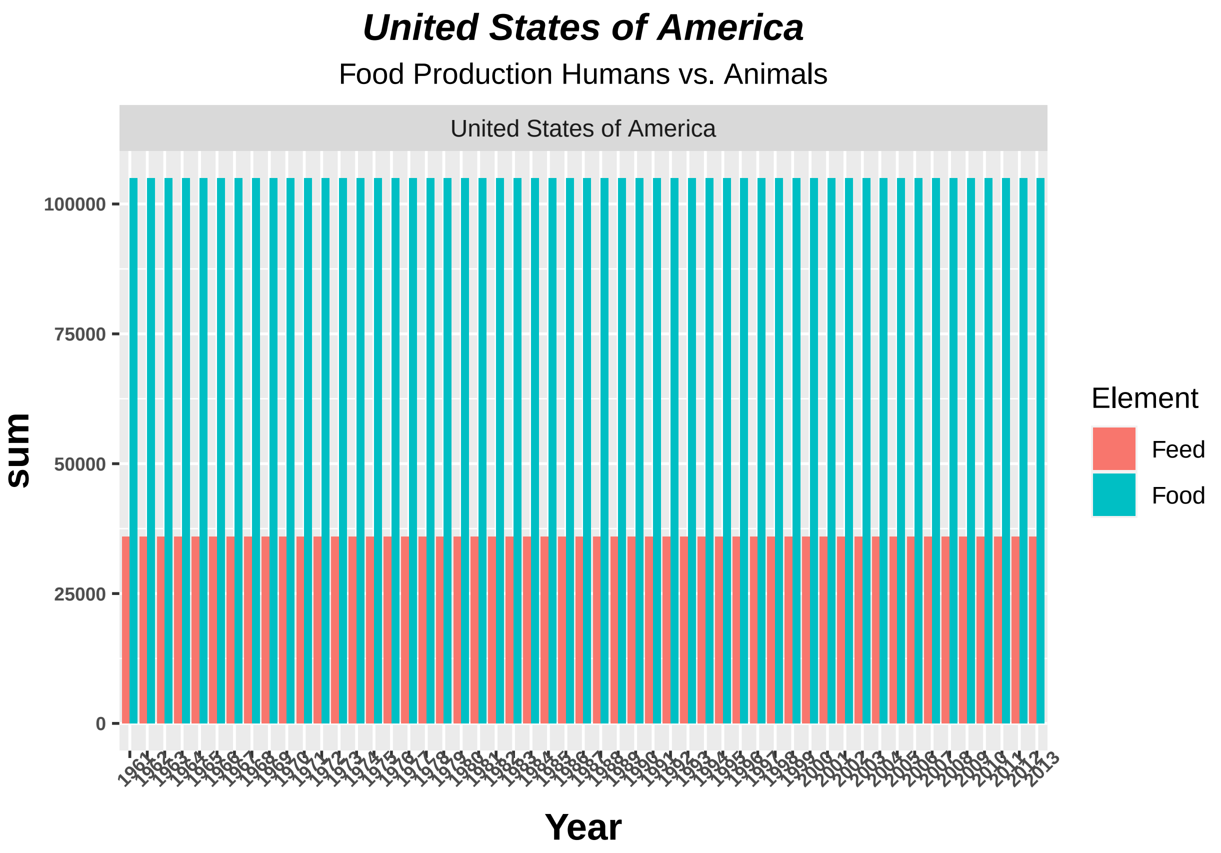
```
## [1] 21477    63
```

```
## # A tibble: 6 x 12
##   ‘Area Abbreviat~ ‘Area Code‘ Area  ‘Item Code‘ Item  ‘Element Code‘ Element
##   <chr>                  <dbl> <chr>       <dbl> <chr>          <dbl> <chr>
## 1 AFG                        2 Afgh~        2511 Whea~           5142 Food
## 2 AFG                        2 Afgh~        2805 Rice~           5142 Food
## 3 AFG                        2 Afgh~        2513 Barl~           5521 Feed
## 4 AFG                        2 Afgh~        2513 Barl~           5142 Food
## 5 AFG                        2 Afgh~        2514 Maiz~           5521 Feed
## 6 AFG                        2 Afgh~        2514 Maiz~           5142 Food
## # ... with 5 more variables: Unit <chr>, latitude <dbl>, longitude <dbl>,
## #   Year <chr>, Count <dbl>
```

```
## [1] Food Feed
## Levels: Feed Food
```

```
## [1] 1000 tonnes
## Levels: 1000 tonnes
```

```
## ‘summarise()‘ regrouping output by ’Area’, ’Year’ (override with ‘.groups‘ argument)
```


United States of America
Food Production Humans vs. Animals
United States of America
sum
100000
75000
50000
25000
0
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
Year
Element
Feed
Food