

Tidy Tuesday

Week 34

AUTHOR
Cristian T

PUBLISHED
August 25, 2024

This week we are exploring English Monarchs and Marriages!

this dataset focuses on the names, ages, and marriages of various 'kings' and 'consorts'. the data ranges all the way back to 850 where the details are a bit fuzzy, spanning all the way to current day. names contain special characters; ages & years can be a bit of a regex puzzle as well. additionally, the age of kings and consorts may show quite a bit of an age gap.

variable	class	description
king_name	character	male or female ruler
king_age	character	ruler's age
consort_name	character	consort chosen to marry king
consort_age	character	age of consort
year_of_marriage	character	historical year of marriage

```
# Load the tidyuesday package
suppressMessages(library(tidyuesdayR)) # For accessing TidyTuesday datasets
suppressMessages(library(skimr)) # For summary and descriptive statistics
suppressMessages(library(tidyverse)) # For data manipulation and visualization
suppressMessages(library(dplyr)) # For data manipulation and transformation
suppressMessages(library(ggplot2)) # For data visualization
suppressMessages(library(RColorBrewer)) # For color palettes in visualizations

# Load the current week's dataset
tuesdata <- tidyuesdayR::tt_load('2024-08-20')
```

Downloading file 1 of 1: `english_monarchs_marriages_df.csv`

```
english_monarchs_marriages_df <- tuesdata$english_monarchs_marriages_df

monarchs <- english_monarchs_marriages_df

# Explore the structure of the dataset
str(monarchs)
```

```
spc_tbl_ [83 × 5] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ king_name      : chr [1:83] "Æthelwulf" "Æthelwulf" "Æthelbald" "Æthelberht" ...
 $ king_age       : chr [1:83] "?" "50(?)" "24" "-" ...
```

```
$ consort_name      : chr [1:83] "Osburh" "Judith of Flanders" "Judith of Flanders" "-" ...
$ consort_age       : chr [1:83] "?" "12" "14" "-" ...
$ year_of_marriage: chr [1:83] "851(?)" "856" "858" "-" ...
- attr(*, "spec")=
  .. cols(
  ..   king_name = col_character(),
  ..   king_age = col_character(),
  ..   consort_name = col_character(),
  ..   consort_age = col_character(),
  ..   year_of_marriage = col_character()
  .. )
- attr(*, "problems")=<externalptr>
```

`skim(monarchs)`

Name	monarchs
Number of rows	83
Number of columns	5
Column type frequency:	
character	5
Group variables	
None	

Data summary

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
king_name	0	1.00	4	22	0	62	0
king_age	6	0.93	1	5	0	33	0
consort_name	0	1.00	1	44	0	71	0
consort_age	0	1.00	1	2	0	27	0
year_of_marriage	0	1.00	1	7	0	70	0

```
# View the first few rows of the dataset
head(monarchs)
```

king_name <chr>	king_age <chr>	consort_name <chr>	consort_age <chr>	
Æthelwulf	?	Osburh	?	
Æthelwulf	50(?)	Judith of Flanders	12	
Æthelbald	24	Judith of Flanders	14	
Æthelberht	–	–	–	
Æthelred	?	Wulfthryth?	?	
Alfred the Great	19	Ealhswith	16	

```
#tidytuesdayR::use_tidytemplate()
```

Clean the data

```
# Replace special characters and uncertainties, convert to numeric where appropriate
monarchs_clean <- monarchs %>%
  mutate(king_age = str_replace_all(king_age, "[—\\?()]", ""), king_age = as.numeric(king_age),
  mutate(consort_age = str_replace_all(consort_age, "[—\\?]", ""), consort_age = as.numeric(consort_age),
  mutate(consort_name = str_replace_all(consort_name, "[—\\?]", ""), consort_name = as.numeric(consort_name),
  mutate(year_of_marriage = str_replace_all(year_of_marriage, "[—\\?()]", ""), year_of_marriage = as.numeric(year_of_marriage))

# Calculate the age gap
monarchs_clean <- monarchs_clean %>%
  mutate(age_gap = king_age - consort_age)

monarchs <- monarchs_clean
```

Missing Values

king_age: 17 missing values (about 0.205 of the rows) consort_age: 25 missing values (about 0.301 of the rows) year_of_marriage: 13 missing values (about 0.157 of the rows) age_gap: 25 missing values (about 0.301 of the rows)

```
# Remove missing values – based on the timeframes, I am not able to validate the accuracy of the data
monarchs <- monarchs %>% filter(!is.na(king_age)) %>% filter(!is.na(consort_age))

# Export Monarchs DF
write.csv(monarchs, "monarchs.csv", row.names = FALSE)
```

Plot 1

Looking at our cleaned data, we have a lot of similar king names, what has been the most popular king name in a thousand years of history?

```
# Find distinct king names and then extract the first name and count occurrences
monarch_names <- monarchs %>% distinct(king_name, .keep_all = TRUE) %>%
  mutate(first_name = word(king_name, 1)) %>%
  count(first_name) %>%
  filter(n >= 2) %>%
  arrange(desc(n))

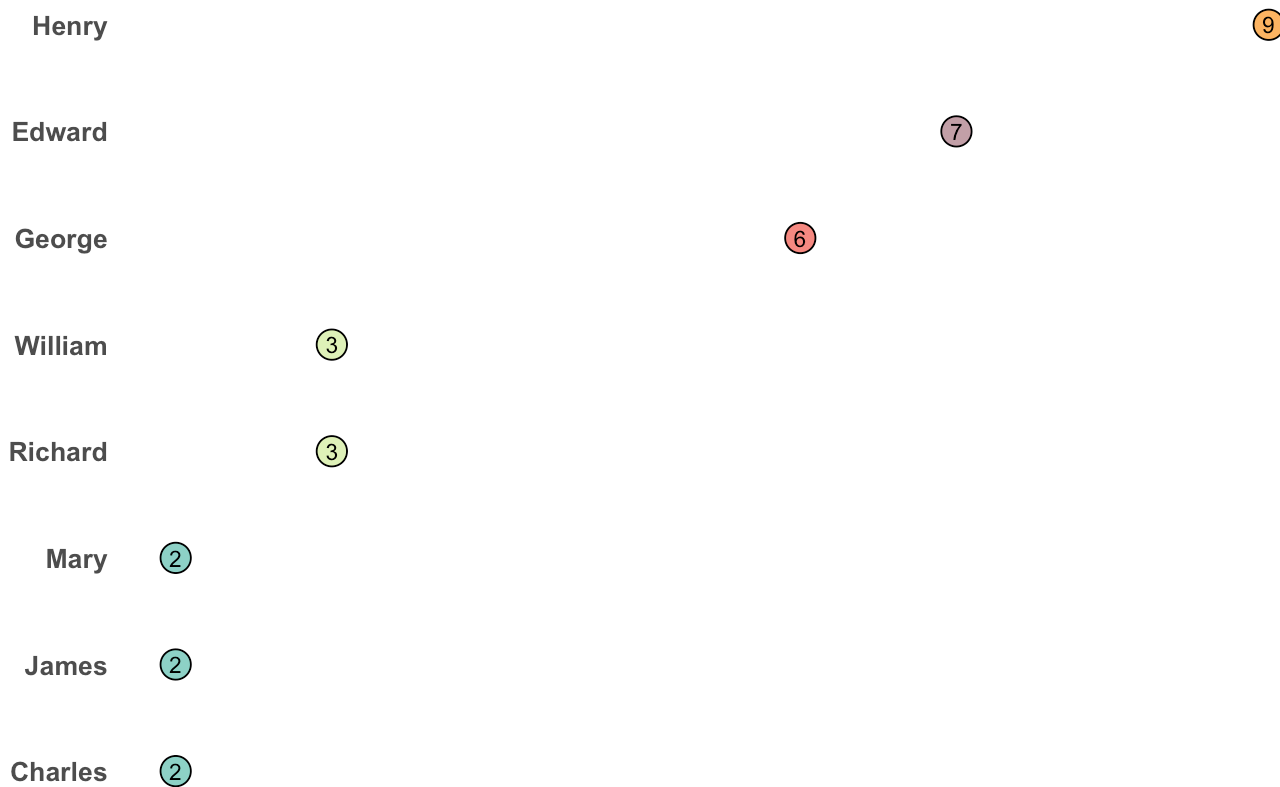
# Plot the results
p <- ggplot(monarch_names, aes(x = n, y = reorder(first_name, n), fill = n)) +
  geom_point(shape = 21, size = 5, color = "black") +
  geom_text(aes(label = n), size = 3, color = "black", vjust = 0.5, hjust = 0.5) +
  scale_fill_gradientn(colors = brewer.pal(6, "Set3")) +
  labs(x = NULL, y = NULL, title = "Monarch Name Popularity") +
  theme_minimal() +
  theme(
```

```

axis.text.y = element_text(size = 10, face = "bold"),
axis.text.x = element_blank(), # Remove x-axis labels
plot.title = element_text(size = 14, face = "bold", hjust = 0.5),
panel.grid.major = element_blank(),
panel.grid.minor = element_blank(),
legend.position = "none"
)
p

```

Monarch Name Popularity



```

#save copy of image
#ggsave("Monarch Name Popularity.png", plot = p, width = 8, height = 5, units = "in")

```

Henry is the Winner with Henry I - VIII (age 14 - 53) and Henry the Young King aged 3

Plot 2

Now what has been the most popular Queen Consort or Prince Consort names?

```

consort_name <- monarchs %>% distinct(consort_name, .keep_all = TRUE) %>%
  mutate(first_name = word(consort_name, 1)) %>%
  count(first_name) %>%
  filter(n >= 2) %>%
  arrange(desc(n))

# Plot the results
p <- ggplot(consort_name, aes(x = n, y = reorder(first_name, n), fill = n)) +
  geom_point(shape = 21, size = 5, color = "black") +

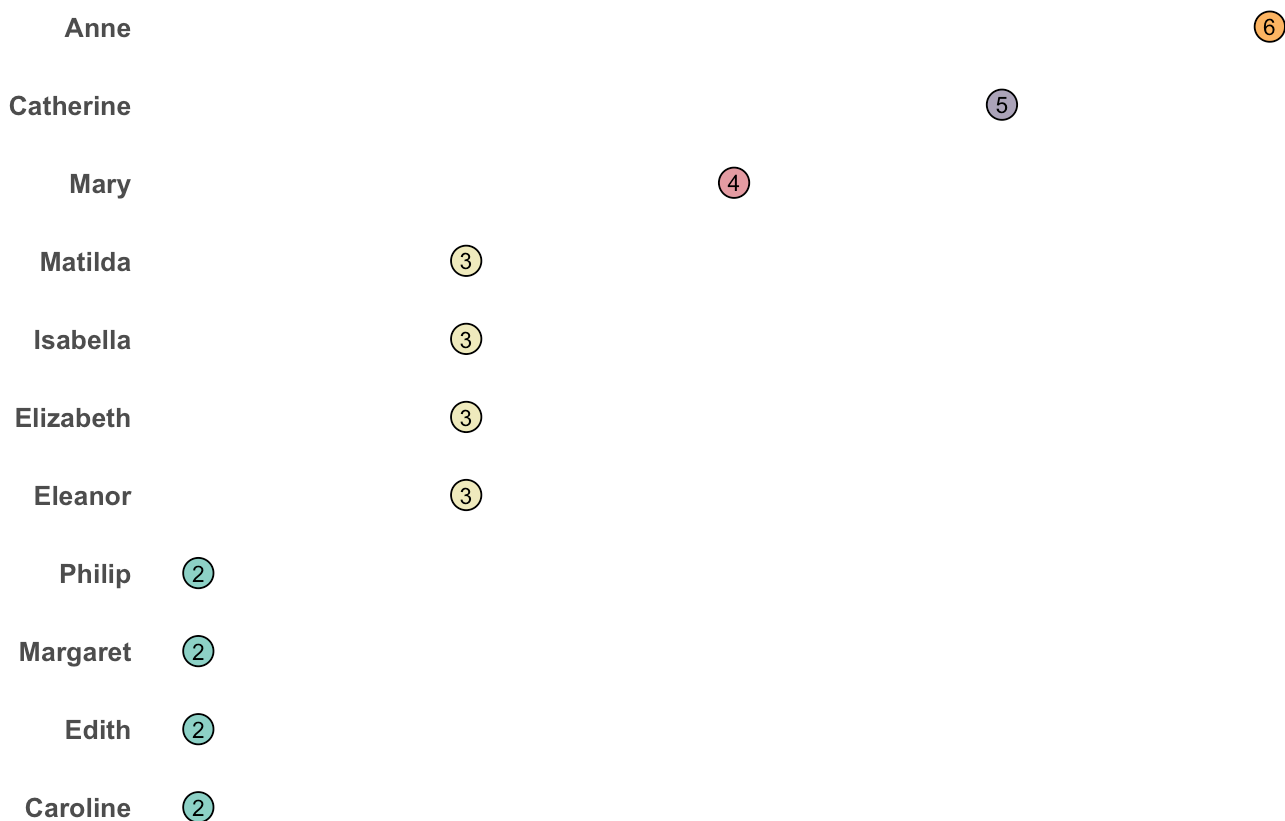
```

```

geom_text(aes(label = n), size = 3, color = "black", vjust = 0.5, hjust = 0.5) +
scale_fill_gradientn(colors = brewer.pal(6, "Set3")) +
labs(x = NULL, y = NULL, title = "Consort Name Popularity") +
theme_minimal() +
theme(
  axis.text.y = element_text(size = 10, face = "bold"),
  axis.text.x = element_blank(), # Remove x-axis labels
  plot.title = element_text(size = 14, face = "bold", hjust = 0.5),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  legend.position = "none"
)
p

```

Consort Name Popularity



```

#save copy of image
#ggsave("Consort Name Popularity.png", plot = p, width = 8, height = 5, units = "in")

```

Anne is the top married consort name and we had 16 year old Anne of Bohemia married in 1382 16 year old Anne Neville married in 1472 32 year old Anne Boleyn married in 1533 25 year old Anne of Cleves married in 1540

15 year old Anne of Denmark married in 1589 and 22 year old Anne Hyde married in 1660

Plot 3

If Anne is the most popular, do Henry's wed Anne's the most?

```

henry_consorts_count <- monarchs %>%
  distinct(consort_name, .keep_all = TRUE) %>%
  mutate(first_name = word(king_name, 1)) %>%
  filter(first_name == "Henry") %>%
  mutate(consort_first_name = word(consort_name, 1)) %>%
  count(consort_first_name) %>%
  filter(n >= 2) %>%
  arrange(desc(n))

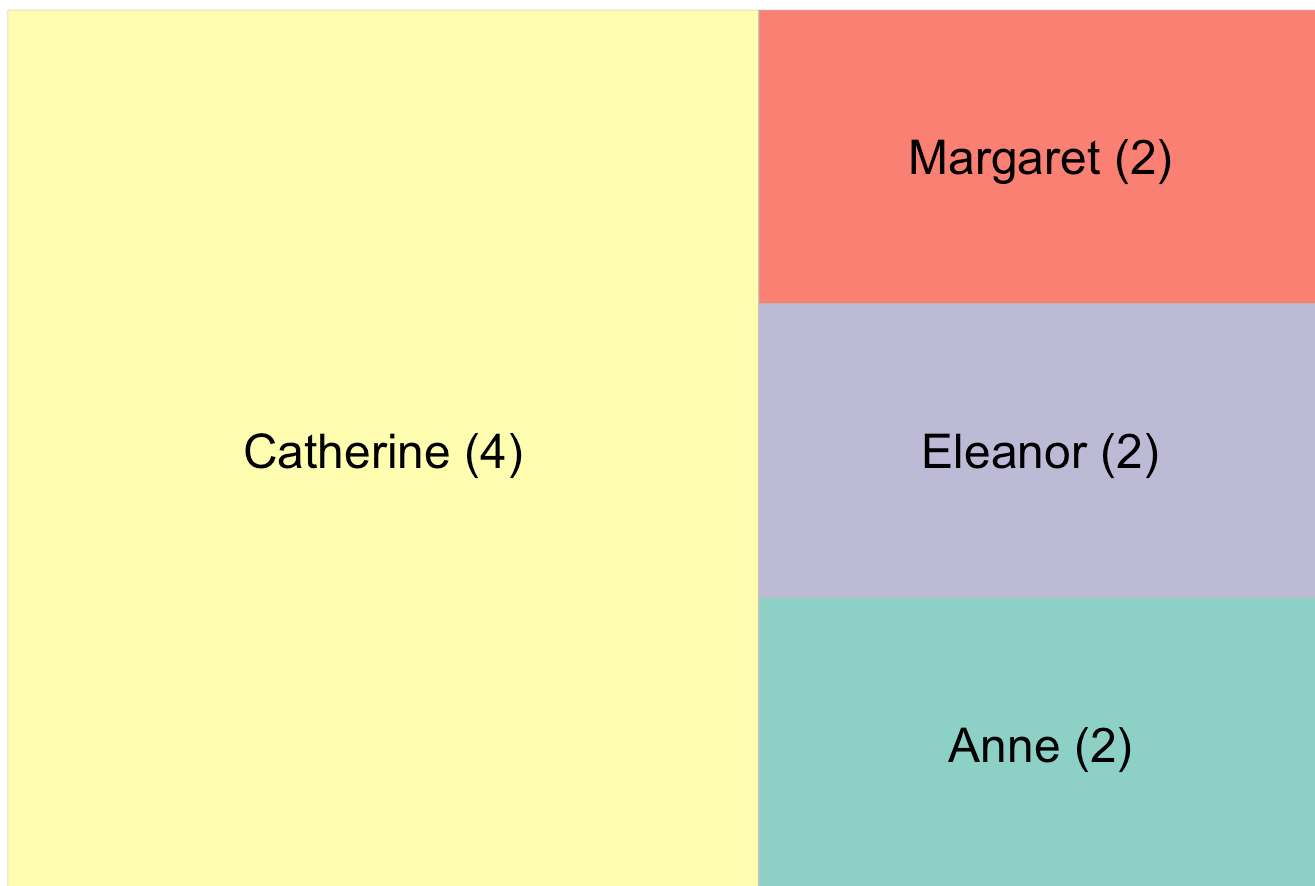
suppressMessages(library(treemapify)) # Extension of ggplot2 to create treemaps

# Manipulate the graph to force treemapify to display a specific way
henry_consorts_count_mod <- data.frame(
  consort_first_name = c("Catherine (4)", "Anne (2)", "Eleanor (2)", "Margaret (2)"),
  n = c(4, 1, 1, 1)
)

# Using Treemap to graph names
ggplot(henry_consorts_count_mod, aes(area = n, fill = consort_first_name, label = consort_first_name)) +
  geom_treemap(aes()) +
  geom_treemap_text(aes(label = consort_first_name), color = "black", place = "centre") +
  scale_fill_brewer(palette = "Set3") +
  labs(title = "Stuck with Henry") +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 14, face = "bold", hjust = 0.5),
    legend.position = "none" # Remove legend for clarity
  )

```

Stuck with Henry



```
#save copy of image  
#ggsave("Stuck with Henry.png", plot = p, width = 8, height = 5, units = "in")
```

They do not, Catherine is the choice among Henry V and Henry VIII. He married 3 different Catherine's

Plot 4

Looking at what may be the most popular graph, what is the age gap between Monarchs and their Consorts?

```
# Create a list of monarchs  
monarch_list <- monarchs %>%  
  select(king_name, year_of_marriage, age_gap)  
  
# Group the data by monarch  
monarch_stats <- monarch_list %>%  
  group_by(king_name) %>%  
  summarise(year_of_marriage = min(year_of_marriage), median_age_gap = median(age_gap))  
  
# Arrange the data by year_of_marriage  
monarch_stats <- monarch_stats %>%  
  arrange(year_of_marriage) %>%  
  mutate(king_name = factor(king_name, levels = rev(unique(king_name))))  
  
# Create a flipped lollipop chart  
  
p <- ggplot(monarch_stats, aes(x = king_name, y = median_age_gap, color = year_of_marriage))  
  geom_segment(aes(x = king_name, xend = king_name, y = 0, yend = median_age_gap), color = "black")  
  geom_point(size = 6) +
```

```

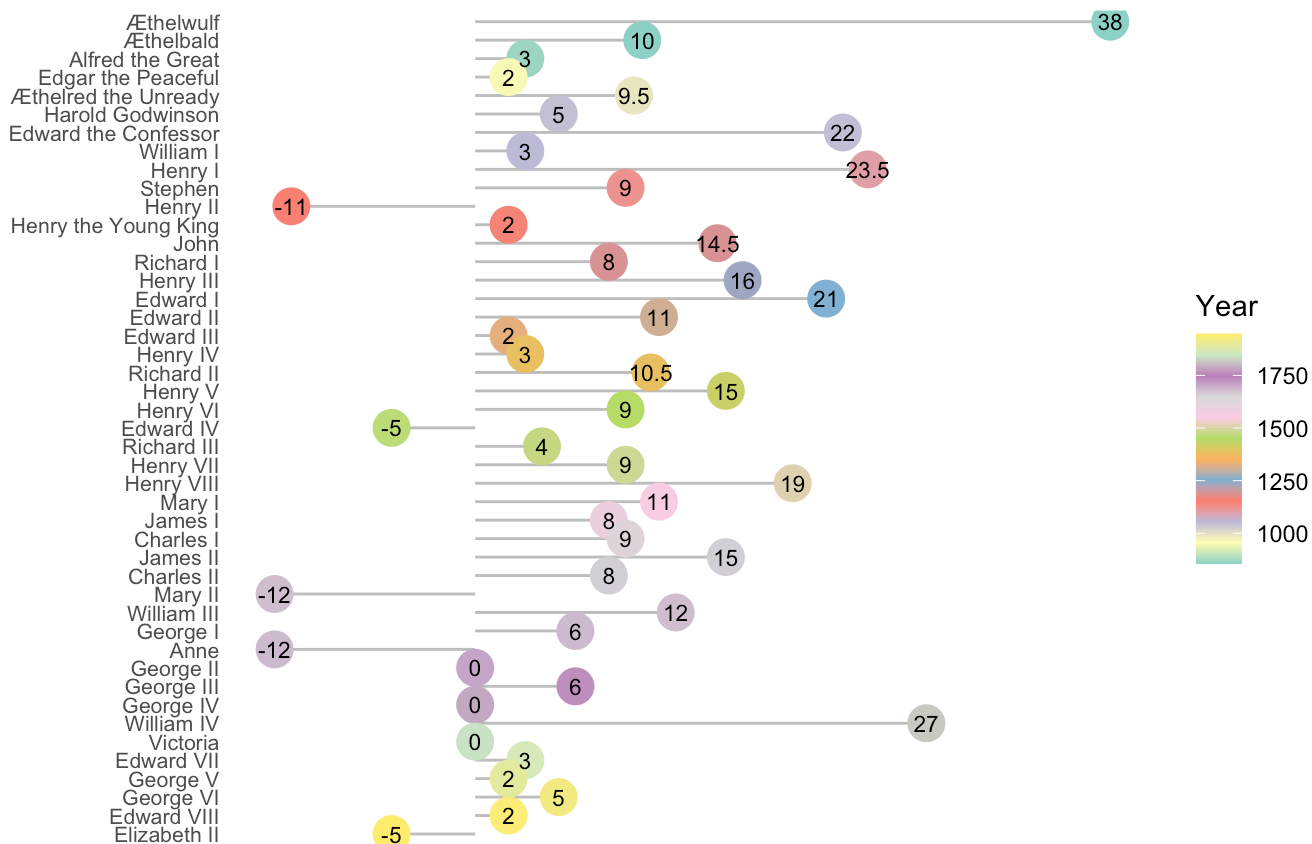
geom_text(aes(label = median_age_gap), vjust = 0.5, hjust = 0.5, color = "black", size = 10) +
scale_color_gradientn(colors = brewer.pal(12, "Set3")) + # Use a color gradient
coord_flip() + # Flip coordinates
theme_minimal() +
labs(title = "Political and Social Norms",
      subtitle = "Age Gap Between Monarchs and Consorts",
      x = NULL,
      y = NULL,
      color = "Year") +
theme(axis.text.y = element_text(size = 8),
      axis.text.x = element_text(size = 10),
      axis.title.x = element_text(size = 12),
      axis.title.y = element_text(size = 12),
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank() #, legend.position = "none")
) +
scale_y_continuous(labels = NULL)

```

p

Political and Social Norms

Age Gap Between Monarchs and Consorts



#save copy of image

```
ggsave("Political and Social Norms.png", plot = p, width = 8, height = 5, units = "in")
```

The graph shows some significant historical shifts in marriage age gaps among monarchs. Early monarchs often had significant median age gaps, with some consorts being much younger, such as Æthelwulf, with a gap of 38 years. Over time, this practice shifted, with negative age gaps appearing as consorts were sometimes older than the monarchs, as seen with Henry II and Mary II. In recent centuries, the median age gaps have generally decreased, reflecting evolving societal norms toward more equal age relationships in

marriage. This trend is evident in more recent monarchs, such as Victoria and Elizabeth II, with much smaller negative gaps.

```
#sessioninfo::session_info(include_base = TRUE)
```

I was curious after the fact to see how names may have changed over history. I created a new column called period broken into before 1000 which encompasses the end of the early medieval period and the start of the high medieval period. 1000-1499 covers the high to late medieval period and the early Renaissance. 1500-1799 includes the Renaissance, Reformation, and Enlightenment periods. 1800 and Later covers the 19th and 20th centuries

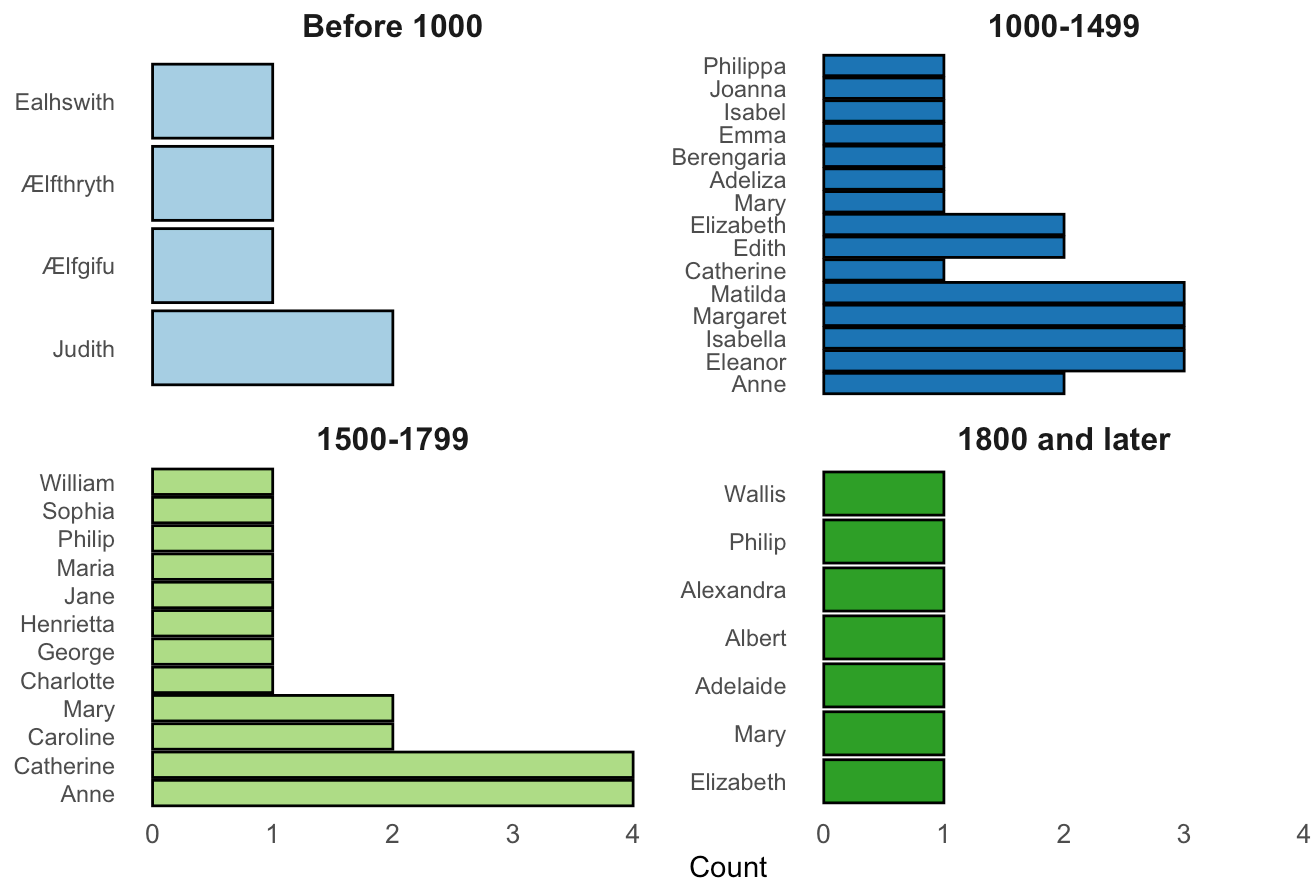
```
# Create the dataset with periods
monarchs_range <- monarchs %>%
  mutate(period = case_when(
    year_of_marriage < 1000 ~ "Before 1000",
    year_of_marriage < 1500 ~ "1000-1499",
    year_of_marriage < 1800 ~ "1500-1799",
    TRUE ~ "1800 and later"
  )) %>%
  mutate(period = factor(period, levels = c("Before 1000", "1000-1499", "1500-1799",

# Group together names by period
consort_first_names_period <- monarchs_range %>%
  mutate(first_name = word(consort_name, 1)) %>%
  count(first_name, period) %>%
  filter(first_name != "") %>%
  arrange(period, desc(n))

# Create the plot with facets
p <- ggplot(consort_first_names_period, aes(x = reorder(first_name, -n), y = n, fill
geom_bar(stat = "identity", color = "black") +
theme_minimal() +
labs(title = "Names of Consorts Over Time",
      x = NULL,
      y = "Count") +
theme(
  axis.text.y = element_text(angle = 0, hjust = 1), # Rotate y-axis labels if need
  axis.text.x = element_text(size = 10),
  strip.text = element_text(face = "bold", size = 12),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  panel.background = element_blank(),
  legend.position = "none"
) +
facet_wrap(~ period, scales = "free_y", ncol = 2, labeller = labeller(period = c(
  "Before 1000" = "Before 1000",
  "1000-1499" = "1000-1499",
  "1500-1799" = "1500-1799",
  "1800 and later" = "1800 and later"
)))) +
scale_fill_brewer(palette = "Paired") +
coord_flip()
```

p

Names of Consorts Over Time



```
#save copy of image
ggsave("First Names of Consorts Over Time.png",plot = p, width = 11, height = 6, unit
```