

Analysis of Vettius Valens

Definitions

```
library(ggplot2)
library(dplyr)

##
## 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
library(cowplot)

##
## *****
## Note: As of version 1.0.0, cowplot does not change the
##   default ggplot2 theme anymore. To recover the previous
##   behavior, execute:
##   theme_set(theme_cowplot())
## *****

# Order of bodies
ORDER_OF_BODIES = c('Saturn',
                    'Jupiter',
                    'Mars',
                    'Sun',
                    'Venus',
                    'Mercury',
                    'Moon')
names(ORDER_OF_BODIES) <- c("Sa", "J", "Mar", "Su", "V", "Mer", "Moo")

# For plots
theme_basic <- function () {
  theme_bw(base_size=12) %>%replace%
  theme(
    axis.text=element_text(colour="black")
  ) %>%replace%
  theme(
    panel.grid=element_blank()
  )
}
```

Singles

We read in the sentiments of the singles.

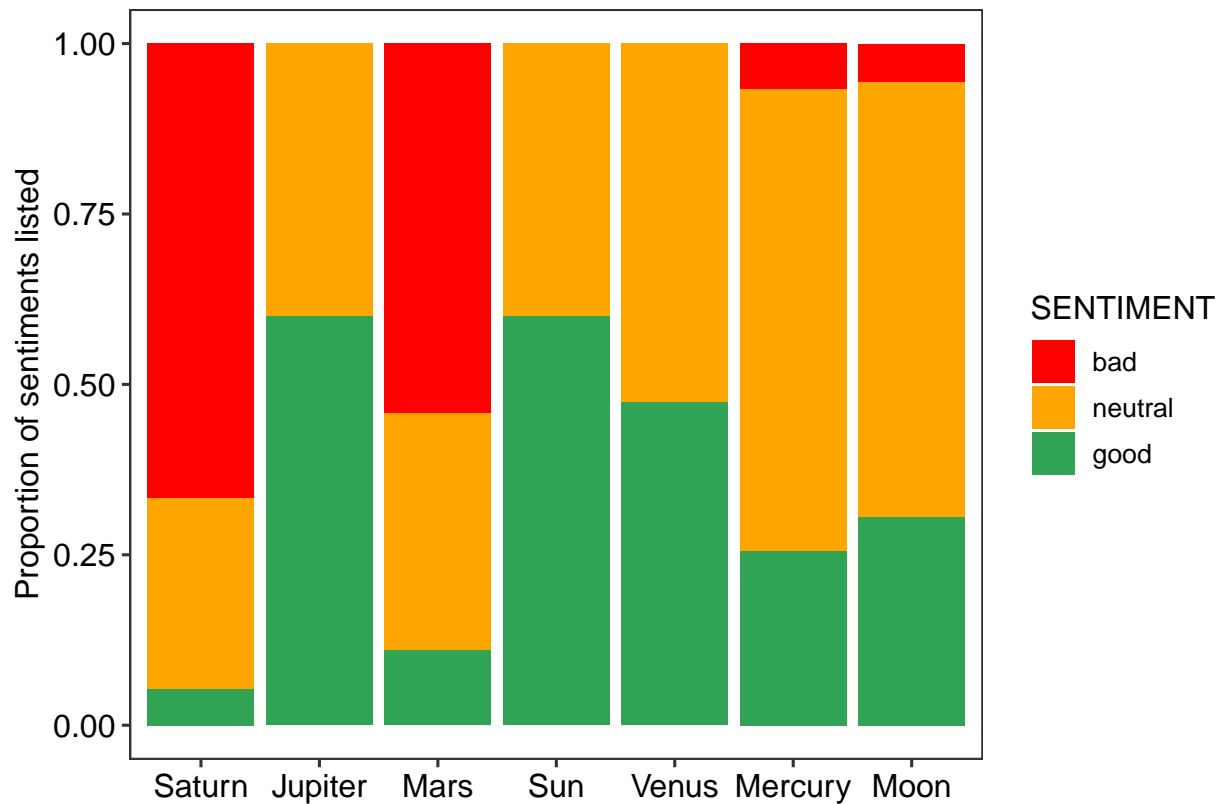
```
singles.sentiments <- read.csv('../data/singles-qualities.csv',
                                header=T,
                                stringsAsFactors = F)

# Get in useful format
singles.sentiments.df <- singles.sentiments %>% group_by(PLANET, SENTIMENT) %>%
  summarise(count=n()) %>%
  mutate(total=sum(count),
         prop=count/total)

# Order planets and sentiments
singles.sentiments.df$PLANET <- ordered(singles.sentiments.df$PLANET,
                                       levels=ORDER_OF_BODIES)
singles.sentiments.df$SENTIMENT <- ordered(singles.sentiments.df$SENTIMENT,
                                           levels=c("bad", "neutral", "good"))
```

We then plot the sentiments of the singles.

```
# Plot them
ggplot(singles.sentiments.df, aes(PLANET, prop, fill=SENTIMENT))+
  geom_bar(stat="identity")+
  theme_basic()+
  xlab("")+
  scale_fill_manual(values=c("red", "orange", "#31a354"))+
  ylab("Proportion of sentiments listed")
```



Doubles

We read in the sentiments of the double conjunctions (e.g. Saturn and Jupiter).

```
doubles.sentiments <- read.csv('../data/doubles-qualities.csv',
                                header=T,
                                stringsAsFactors = F)

# Get in useful format
doubles.sentiments.df <- doubles.sentiments %>% group_by(DOUBLE, SENTIMENT) %>%
  summarise(count=n()) %>%
  mutate(total=sum(count),
         prop=count/total)

# Order planets and sentiments
doubles.sentiments.df$body.1 <- sapply(stringr::str_split(doubles.sentiments.df$DOUBLE, pattern=" "),
                                       function(x) x[1])
doubles.sentiments.df$body.1 <- ordered(ORDER_OF_BODIES[doubles.sentiments.df$body.1],
                                       levels=ORDER_OF_BODIES)
doubles.sentiments.df$body.2 <- sapply(stringr::str_split(doubles.sentiments.df$DOUBLE, pattern=" "),
                                       function(x) x[2])
doubles.sentiments.df$body.2 <- ordered(ORDER_OF_BODIES[doubles.sentiments.df$body.2],
                                       levels=ORDER_OF_BODIES)
doubles.sentiments.df$bodies.sorted <- sapply(1:nrow(doubles.sentiments.df),
                                             function(x)
                                              paste(as.character(sort(unlist(c(doubles.sentiments.df[x,"body.1"],
                                                                                   doubles.sentiments.df[x,"body.2"]
                                                                                   collapse=" ")))

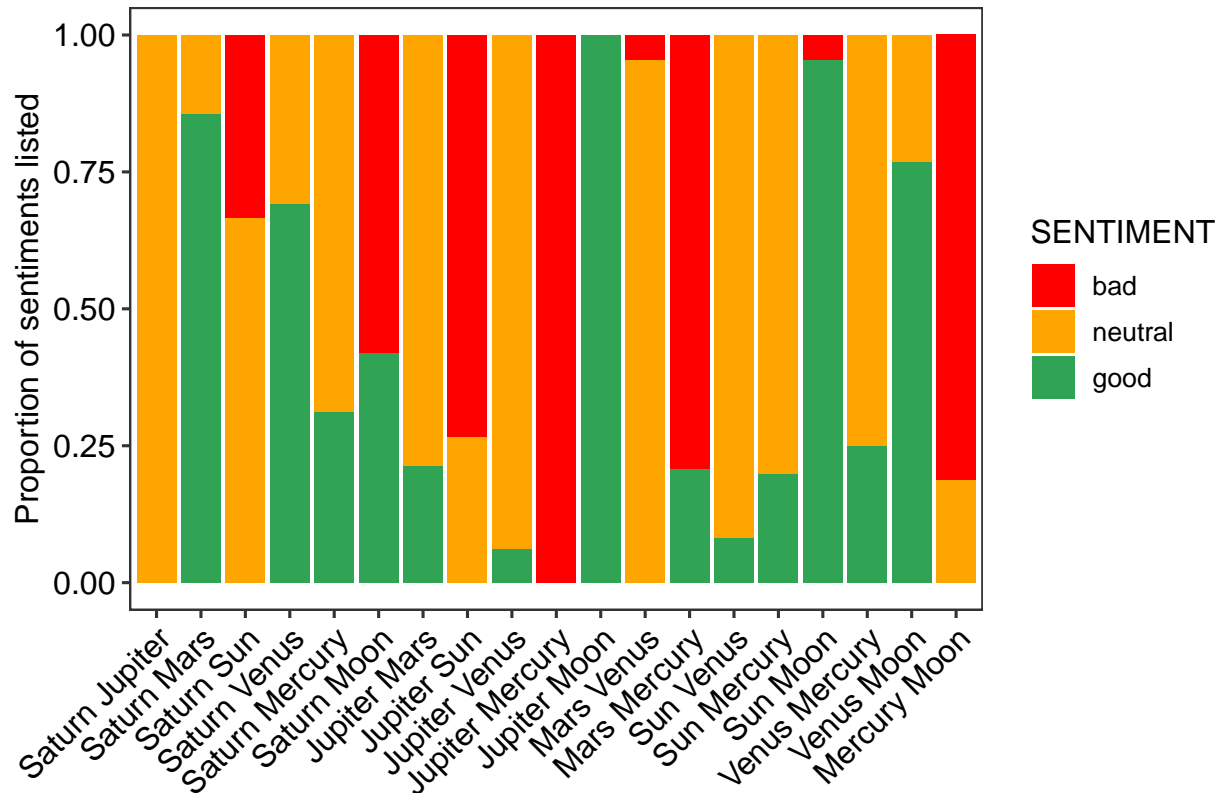
# Make body 1 and 2 be in order of bodies as expected
doubles.sentiments.df$body.1 <- gsub(".*", "", doubles.sentiments.df$bodies.sorted)
doubles.sentiments.df$body.2 <- gsub(".* ", "", doubles.sentiments.df$bodies.sorted)
doubles.sentiments.df$body.1 <- ordered(doubles.sentiments.df$body.1,
                                       levels=ORDER_OF_BODIES)
doubles.sentiments.df$body.2 <- ordered(doubles.sentiments.df$body.2,
                                       levels=ORDER_OF_BODIES)
doubles.sentiments.df$order.body.string <- paste0(as.numeric(doubles.sentiments.df$body.1),
                                                as.numeric(doubles.sentiments.df$body.2))

# Order bodies again (this is hacky but it works)
doubles.sentiments.df$bodies.sorted <- ordered(doubles.sentiments.df$bodies.sorted,
                                              levels=unique(doubles.sentiments.df$bodies.sorted[order(d

doubles.sentiments.df$SENTIMENT <- ordered(singles.sentiments.df$SENTIMENT,
                                           levels=c("bad", "neutral", "good"))
```

We then plot the sentiments of the doubles.

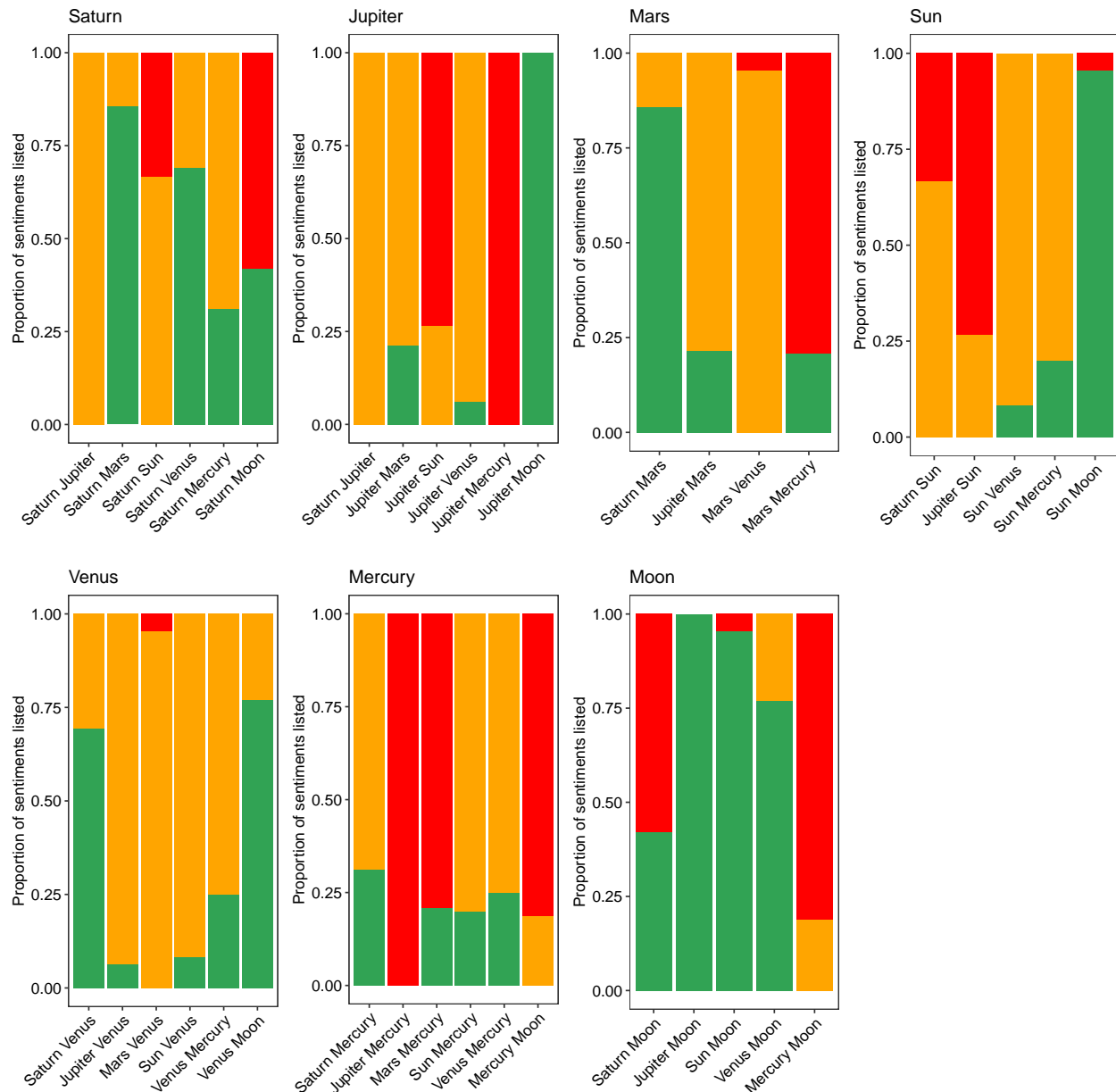
```
# Plot them
ggplot(doubles.sentiments.df, aes(bodies.sorted, prop, fill=SENTIMENT))+
  geom_bar(stat="identity")+
  theme_basic()+
  xlab("")+
  scale_fill_manual(values=c("red", "orange", "#31a354"))+
  ylab("Proportion of sentiments listed")+
  theme(axis.text.x=element_text(angle=45, hjust=1))
```



A function to plot all the doubles involving a particular planet.

```
plotDoublesWithPlanet <- function(planet){
  # Subset to only doubles involving planet
  local.planet.df <- doubles.sentiments.df[grep(planet, doubles.sentiments.df$bodies.sorted),]
  ggplot(local.planet.df, aes(bodies.sorted, prop, fill=SENTIMENT))+
    geom_bar(stat="identity")+
    theme_basic()+
    xlab("")+
    scale_fill_manual(values=c("red", "orange", "#31a354"))+
    ylab("Proportion of sentiments listed")+
    theme(axis.text.x=element_text(angle=45, hjust=1))+
    ggtitle(planet)+
    theme(legend.position = "none")
}

# Make all these plots and combine them
p.doubles.saturn <- plotDoublesWithPlanet("Saturn")
p.doubles.jupiter <- plotDoublesWithPlanet("Jupiter")
p.doubles.mars <- plotDoublesWithPlanet("Mars")
p.doubles.sun <- plotDoublesWithPlanet("Sun")
p.doubles.venus <- plotDoublesWithPlanet("Venus")
p.doubles.mercury <- plotDoublesWithPlanet("Mercury")
p.doubles.moon <- plotDoublesWithPlanet("Moon")
cowplot::plot_grid(p.doubles.saturn, p.doubles.jupiter,
  p.doubles.mars, p.doubles.sun,
  p.doubles.venus, p.doubles.mercury,
  p.doubles.moon, nrow=2)
```



Note that Valens misses out two doubles which involve Mars: Mars + Sun, and Mars + Moon.

Predicting doubles from singles

If we know the sentiment proportions associated with single planets, can we predict the sentiment of the doubles?

```
getSingleScores <- function(double, sentiment="good", mean=TRUE, singles=singles.sentiments.df){
  double.bodies <- ordered(unlist(stringr::str_split(double, pattern=" "),
                             levels=ORDER_OF_BODIES)

  prop <- 0
  for (body in double.bodies){
    new.prop <- as.numeric(singles[which(singles$PLANET==body &
                                         singles$SENTIMENT==sentiment), "prop"] )

    prop <- prop + new.prop
  }
}
```

```

    }
    if (mean==FALSE){
      return(prop)
    }
    else{
      return(prop/2)
    }
  }
doubles.sentiments.df.good <- doubles.sentiments.df[which(doubles.sentiments.df$SENTIMENT=="good"),]

doubles.sentiments.df.good$mean.single.sentiments <- sapply(doubles.sentiments.df.good$bodies.sorted,
  function(x) getSingleScores(x, sentiment = "good", mean=TRUE))

p.good <- ggplot(doubles.sentiments.df.good, aes(mean.single.sentiments, prop))+
  geom_point(colour="#31a354")+
  theme_basic()+
  geom_text(aes(label=bodies.sorted))+
  coord_fixed()+
  xlim(c(0,1))+
  ylim(c(0,1))+
  ggtitle("Good")

# Check for neutral as well
doubles.sentiments.df.neutral <- doubles.sentiments.df[which(doubles.sentiments.df$SENTIMENT=="neutral"),]

doubles.sentiments.df.neutral$mean.single.sentiments <- sapply(doubles.sentiments.df.neutral$bodies.sorted,
  function(x) getSingleScores(x, "neutral", mean=TRUE))

p.neutral <- ggplot(doubles.sentiments.df.neutral, aes(mean.single.sentiments, prop))+
  geom_point(colour="orange")+
  theme_basic()+
  ggrepel::geom_text_repel(aes(label=bodies.sorted), nudge_x = 0.02, nudge_y=-0.025)+
  coord_fixed()+
  xlim(c(0,1))+
  ylim(c(0,1))+
  ggtitle("Neutral")

# And bad
doubles.sentiments.df.bad <- doubles.sentiments.df[which(doubles.sentiments.df$SENTIMENT=="bad"),]

doubles.sentiments.df.bad$mean.single.sentiments <- sapply(doubles.sentiments.df.bad$bodies.sorted,
  function(x) getSingleScores(x, "bad", mean=TRUE))

p.bad <- ggplot(doubles.sentiments.df.bad, aes(mean.single.sentiments, prop))+
  geom_point(colour="red")+
  theme_basic()+
  ggrepel::geom_text_repel(aes(label=bodies.sorted), nudge_x = 0.02, nudge_y=-0.025)+
  coord_fixed()+
  xlim(c(0,1))+
  ylim(c(0,1))+
  ggtitle("Bad")

cowplot::plot_grid(p.good, p.neutral, p.bad, nrow=3)

```

```
## Warning: Removed 5 rows containing missing values (geom_point).
```

```
## Warning: Removed 5 rows containing missing values (geom_text_repel).
```

