Anton_documentation

Anton Hung

2022-11-15

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
setwd('/Volumes/GoogleDrive/Mon disque/wrangling/project/wranglinghub')

data <- read_csv('football_data.csv')
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
## Rows: 13504 Columns: 36
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (17): schedule_date, schedule_week, team_home, Home team abbrev, team_aw...
## dbl (17): schedule_season, score_home, score_away, spread_favorite, over_und...
## lgl  (2): schedule_playoff, stadium_neutral
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(data)
```

```
## # A tibble: 6 x 36
##   schedule_date schedu~1 sched~2 sched~3 team_~4 Home ~5 score~6 score~7 team_~8
##   <chr>            <dbl> <chr>   <lgl>   <chr>   <chr>     <dbl>   <dbl> <chr>
## 1 9/2/1966          1966 1       FALSE   Miami ~ MIA          14      23 Oaklan~
## 2 9/3/1966          1966 1       FALSE   Housto~ TEN          45       7 Denver~
## 3 9/4/1966          1966 1       FALSE   San Di~ LAC          27       7 Buffal~
## 4 9/9/1966          1966 2       FALSE   Miami ~ MIA          14      19 New Yo~
```

```
## 5 9/10/1966         1966 1       FALSE   Green ~ GB           24       3 Baltim~
## 6 9/10/1966         1966 2       FALSE   Housto~ TEN          31       0 Oaklan~
## # ... with 27 more variables: `Away team abbrev` <chr>, team_favorite_id <chr>,
## #   spread_favorite <dbl>, over_under_line <dbl>, stadium <chr>,
## #   stadium_neutral <lgl>, weather_temperature <dbl>, weather_wind_mph <dbl>,
## #   weather_humidity <dbl>, weather_detail <chr>,
## #   Difference_favored_minus_notfavored <dbl>, `Abs value of spread` <dbl>,
## #   `Actual difference - spread` <dbl>, stadium_location <chr>,
## #   stadium_open <dbl>, stadium_close <dbl>, stadium_type <chr>, ...
```

Functions library

For our project, we chose to include a functions library, a separate R script containing all the functions in our code. This is useful because it reduces the amount of copying and pasting of our code. It also makes the document look more presentable when we aren't taking up space with trivial bits of code. Here is a required code chunk for importing the functions from our separate r script:

```
setwd('/Volumes/GoogleDrive/Mon disque/wrangling/project/wranglinghub')
source('functions_library/functions_library.R')
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

Missing data

Almost every dataset will be missing some data. This section will cover how we explored where we are missing data and how we handled it.

The library "naniar" contains many methods for visualizing which columns have missing data. Here, we observed that "weather_detail" has the most missing values This variable contains 8 factors: "DOME",

"DOME (Open Roof)", "Fog", "Rain", "Rain, Fog", "Snow", "Snow, Fog", and "Snow, Freezing Rain". All of these factors are related to poor weather, or dome-related information. Values were not recorded for "nice" weather, which is probably contributing to the large amount of missing values. To proceed, we simply decided not to use this column in our analysis. There were many other weather-related columns available in the dataset.

Later, in the web scraping section of the documentation, we describe how we imputed data into a new column to replace this "weather detail" column.

```
data_missing <- data
library(naniar)
# gg_miss_var(data_missing) # this causes a latex error when I try to knit it

levels(as.factor(data_missing$weather_detail))
```

```
## [1] "DOME"                "DOME (Open Roof)"    "Fog"
## [4] "Rain"                "Rain | Fog"          "Snow"
## [7] "Snow | Fog"          "Snow | Freezing Rain"
```

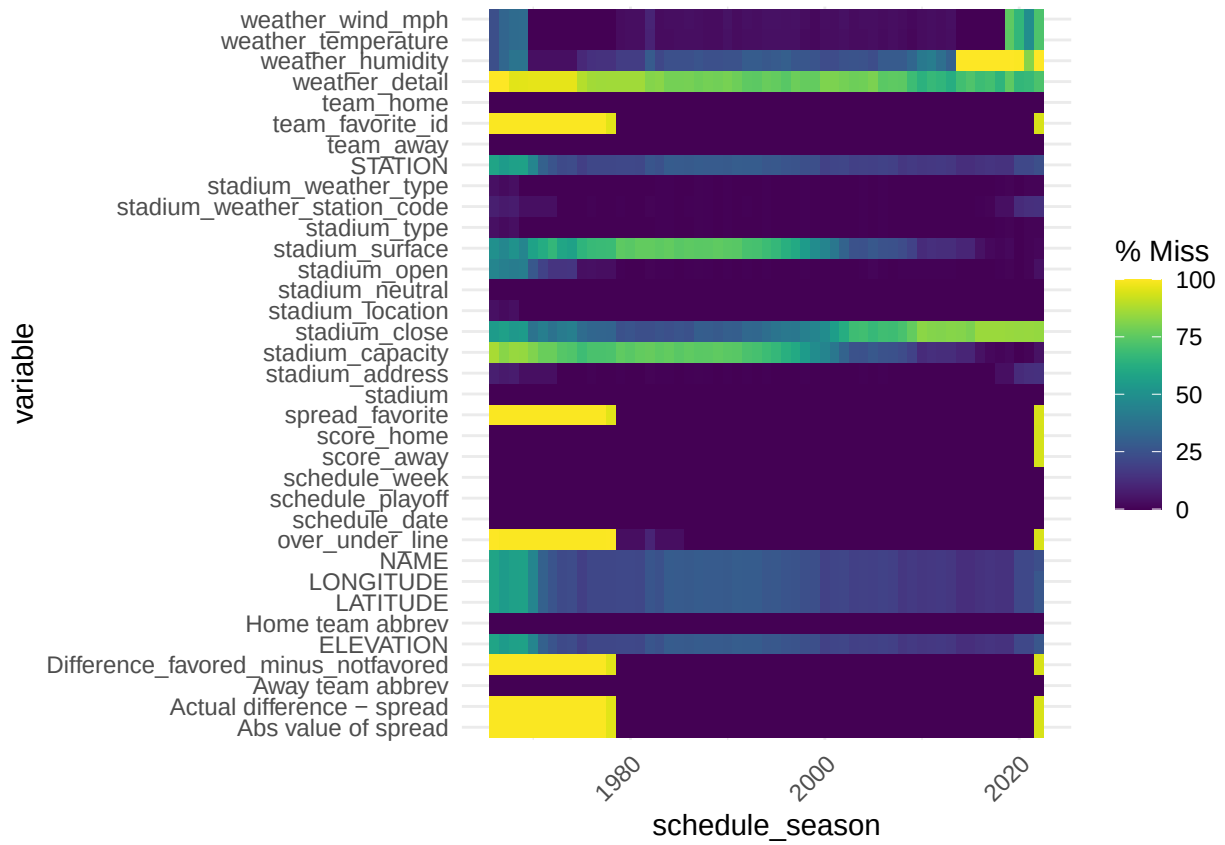
Next, we want to confirm whether our data is missing at random, or if there is a pattern to the missingness. mcar_test from the naniar library returns a value less than 0.05 if our data is not missing at random.

```
library(naniar)

# mcar_test(data.frame(data)) # this runs into an error, but we can subset the data by columns and
# run an mcar test on just a portion of the data

# this confirms that the data is missing not a random
mcar_test(data.frame(data[,10:20])) # p-value = 0
```

```
## # A tibble: 1 x 4
##   statistic    df p.value missing.patterns
##       <dbl> <dbl>   <dbl>            <int>
## 1    10578.   102       0               15
```

Here is another visualization of our missing data, a heatmap. We can see where we can see our data is missing, sorted by date. Evidently, we can confirm that there is a pattern to our missingness. There is a lot of missing data related to betting up until near 1978. Also, there is some missing data in the 2023 season, where the authors of the dataset pre-allotted some rows for games that haven't been played yet.

```
library(naniar)
gg_miss_fct(x = data_missing, fct = schedule_season)
```



Looking closely at specifically our data related to betting, we confirmed that these columns are where we have the majority of our missing values. Columns related to the over/under line, the spread, and the favored team.

```
pct_miss(data) # 14.7%
```

```
## [1] 14.73679
```

```
betting_columns <- c("team_favorite_id",
                     "spread_favorite",
                     "over_under_line",
                     "Difference_favored_minus_notfavored",
                     "Actual difference - spread",
                     "Abs value of spread")
pct_miss(data[,betting_columns]) # 20.34%
```

```
## [1] 20.34212
```

```
pct_miss(data[, -which(colnames(data) %in% betting_columns)]) # 13.6%
```

```
## [1] 13.61572
```

Looking at the data in excel, our regular season betting data begins at row 2494 (2493 if we do not include the header).

```
pct_miss(data[1:2492,betting_columns]) # 99.5% of betting data missing pre-1978
```

```
## [1] 99.49171
```

```
pct_miss(data[2493:13248,betting_columns]) # 11% of betting data missing between 1978 and present
```

```
## [1] 0.1084666
```

```
pct_miss(data[13249:nrow(data),betting_columns]) # 100% of betting data missing in the future
```

```
## [1] 100
```

We have confirmed that between rows 2493-13248 is where we have most of our meaningful data.

Analysis of how our weather data affects the accuracy of the spread.

This section will demonstrate how we analyzed the weather data in our dataset to determine its effect on the accuracy of the spread. First, we subset the original dataset to include only football seasons from 1978-present. Prior to this, we do not have betting data for every game.

```
betting_data <- data[2493:13248,] # keep only a subset of the rows
```

The columns of our dataset that we are interested in for this section are:

```
### Weather columns:
# weather_temperature
# weather_wind_mph
# weather_humidity
# stadium_weather_type

### Stadium Surface:
# stadium_surface

### Outcome (Accuracy of the spread):
# Actual difference - spread
```

Temperature, wind, and humidity

Here, we are simply separating our rows into games played in a domed stadium vs not a domed stadium. This is because our weather variables are held pretty constant inside of a dome, so really we only expected to see variation in games played outside of a dome.

```
# weather_temperature
# weather_wind_mph
# weather_humidity
domed_stadiums <- filter(betting_data,
```

```
                        stadium_weather_type=='dome')

non_domed_stadiums <- filter(betting_data,
                        stadium_weather_type=='cold' |
                          stadium_weather_type=='moderate' |
                          stadium_weather_type=='warm')
```

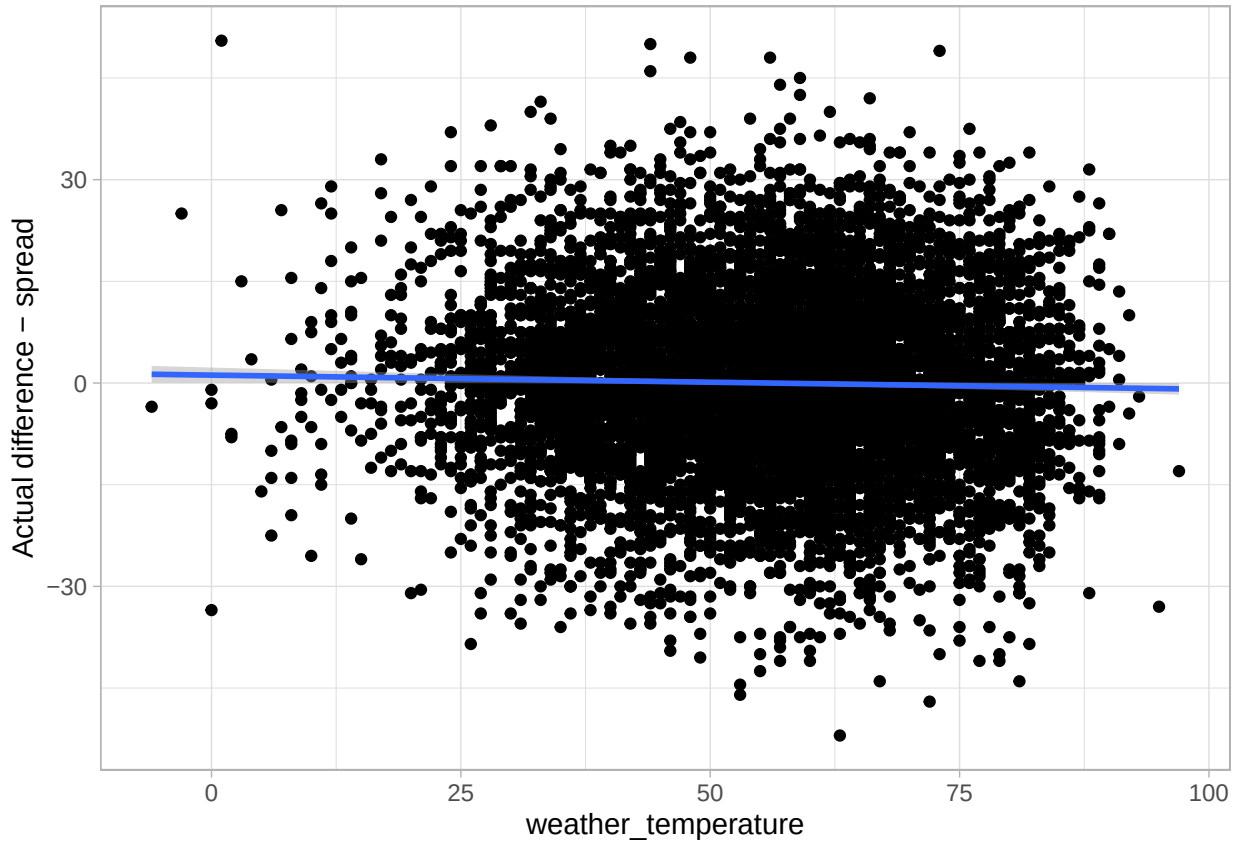
Exploring our weather variables using scatterplots, and identifying correlation.

From the plots, we can see that there is no correlation. The slopes are flat. cor() returns correlation coefficients, where closer to 1 means stronger correlation. Our correlation coefficients are all very close to 0. The conclusion here is that none of these variables are affecting the accuracy of our spread data.

```
ggplot(non_domed_stadiums, aes(x=weather_temperature, y=`Actual difference - spread`)) +
    geom_point()+
    geom_smooth(method='lm', formula = y~x) +
    theme_light()
```

```
## Warning: Removed 799 rows containing non-finite values (‘stat_smooth()‘).
```

```
## Warning: Removed 799 rows containing missing values (‘geom_point()‘).
```



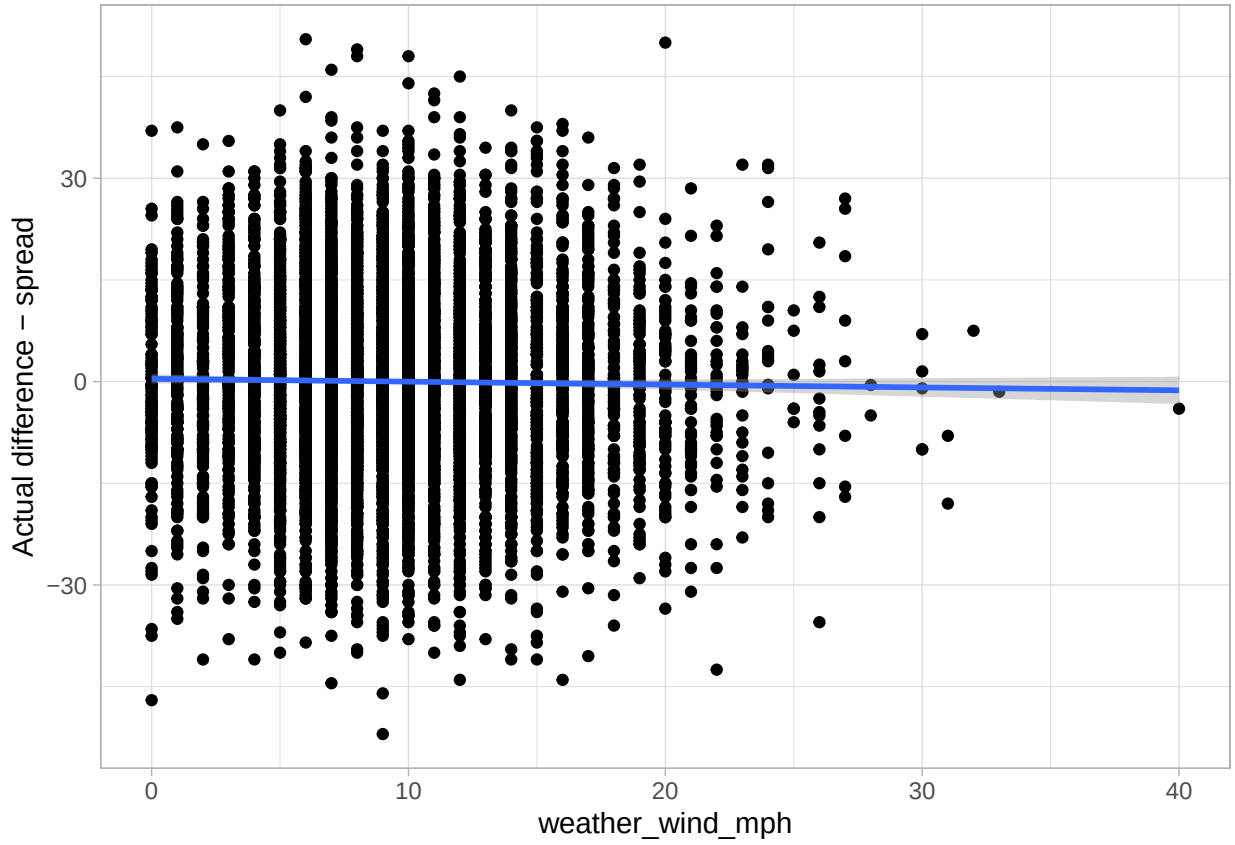
```
cor(non_domed_stadiums$weather_temperature, non_domed_stadiums$`Actual difference - spread`, use= "compl
```

```
## [1] -0.02477375
```

```
ggplot(non_domed_stadiums, aes(x=weather_wind_mph, y=`Actual difference - spread`)) +
    geom_point()+
    geom_smooth(method='lm', formula = y~x) +
    theme_light()
```

```
## Warning: Removed 810 rows containing non-finite values (‘stat_smooth()‘).
```

```
## Warning: Removed 810 rows containing missing values (‘geom_point()‘).
```



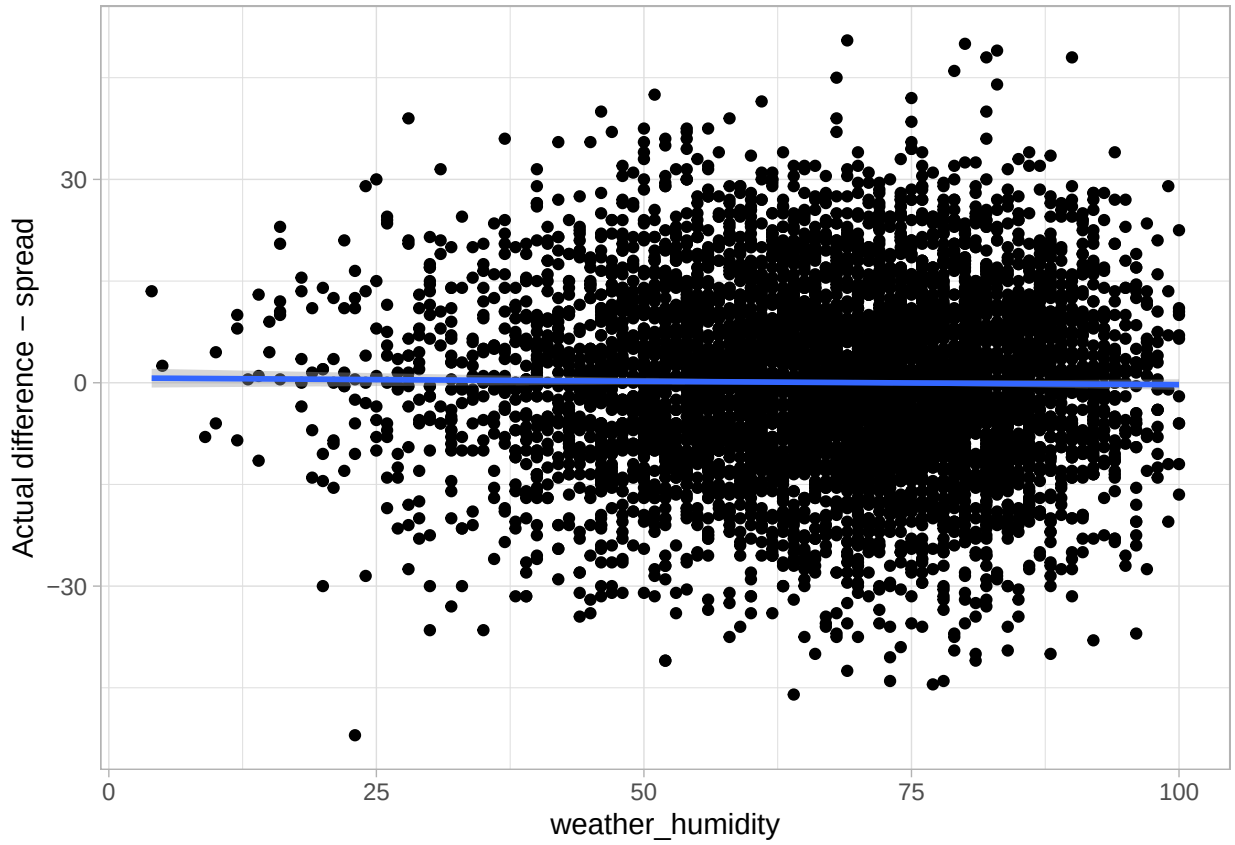

```
cor(non_domed_stadiums$weather_wind_mph, non_domed_stadiums$`Actual difference - spread`, use= "complete
```

```
## [1] -0.01495849
```

```
ggplot(non_domed_stadiums, aes(x=weather_humidity, y=`Actual difference - spread`)) +
    geom_point()+
    geom_smooth(method='lm', formula = y~x) +
    theme_light()
```

```
## Warning: Removed 1921 rows containing non-finite values (‘stat_smooth()‘).
```

```
## Warning: Removed 1921 rows containing missing values (`geom_point()`).
```




```
cor(non_domed_stadiums$weather_humidity, non_domed_stadiums$`Actual difference - spread`, use= "complete
```

```
## [1] -0.01162643
```

Using our weather variables to assign games to categories of different weather conditions

I defined three weather categories: Ideal, Ok, and poor weather. To do this, I determined cutoff values for temperature and wind for each category. These cutoffs were chosen by looking at the 1st and 3rd quartiles of our data. Poor weather was defined as having a wind speed > 12 mph (top 25% windiest days) and a temperature <45 (top 25% coldest days) or >67.5 (top 25% hottest days). Ideal weather was defined as wind below 6 mph, and temperature between 45-67.5. Any game played under a dome was also considered ideal weather. Any games not categorized under poor or ideal weather were then simply grouped into "Ok" weather.

Next, these three categories can be compared to the accuracy of our betting spread using a violin plot. All three weather categories appeared to have nearly identical distributions of spread accuracy.

```
summary(non_domed_stadiums$weather_temperature) # 45, 57, 67.5
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   -6.00   45.00   57.00   55.83   67.50   97.00     799
```

```
summary(non_domed_stadiums$weather_wind_mph) # 6, 9, 12
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   6.000   9.000   9.538  12.000  40.000     810
```

```
summary(non_domed_stadiums$weather_humidity) # 57, 69, 78
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    4.00   57.00   69.00   67.03   78.00  100.00    1921
```

```
plot_weather_status(betting_data)
```