Roller Coasters: Game versus Reality

Edoardo Bianchi

18/01/2022

Roller coasters are the rides that make millions of people visit theme parks every year and it does not exist a theme park without at least one roller coaster. In Italy there are different theme parks with some roller coasters, for example Gardaland, but this project is not about rides in Italy. In fact **i am going to analyze not only a set of real roller coasters present in the US, but also a set of rides created in a popular theme park "simulator": Planet Coaster**. Then i will **compare the two sets of rides** to understand if game rides are realistic and can exist in real life too.

Planet Coaster is a famous computer and console video game developed by Frontier in which the player creates and manages theme parks: from terrain editor and ride building to staff hiring to marketing campaigns. Basically the main goal is make the guests of the park happy, and in order to do so building attractive roller coasters is one of the most important tasks. When the player builds a new ride, this ride is tested by the artificial intelligence of the game, that simulates real riders, and then a final rating is shown. The rating system consists of three metrics: Excitement, Fear and Nausea, all this metrics must be in a certain interval to classify a ride as "good" in the game.

Summarizing, this project has **several tasks**:

- **Analyze the real rides** present in the US from a technical and spatial-temporal perspective (how they are distributed in the country, how they evolved, . . .).
- **Analyze the game created rides** and understand what are the factors that influence the most the rating: how to build a good ride?
- **Compare the game created rides with real life existing rides** and understand the differences: can game rides exist in reality?

At the end we aim to **answer different questions**, for example:

- What are the factors that determines a good ride?
- How the height and speed of rides changed during the years?
- How the number of inversions changed during the years?
- What are the states with the highest number of rides and why?
- What are the differences between real and game created rides, if there is one?
- Can the game created rides exist in real life or game ride are not realistic? And many other.

Steps of the Project

This project consist of the following sections:

1. **Importing Tools**
2. **Data Reading**
3. **Data Cleaning and Preprocessing**
4. **Real Rides Data Visualization**
5. **Game Rides Data Visualization**
6. **Game VS Reality: Roller Coaster Comparison**
7. **Drawing Conclusions - Summary**

Data Sets

For this project we need **three different data sets**:

1. **one for the rides created in the game** –> game rides visualization.
2. **another for the real life existing rides** –> real rides visualization.
3. **the last one is a merge between the (1) and (2)** –> visualize the differences between game and real rides.

The first data set it's been created by myself, and consists of **multiple game created rides**. It contains all the technical specifications of the rides. A new column will be added and will contain the **coaster class**, that is a flag indicating if the ride is good or not good. The value of this flag is calculated on the three metrics used by Planet Coaster to evaluate a ride: Excitement, Nausea and Fear. It is important to note that a ride is classified as "good" when:

1. Excitement is greater than 6.34
2. Nausea is lesser than 3.5
3. Fear is greater than 4.0 and lesser than 6.0

Each row is a roller coaster created in the game, not all the rides are created by me. The majority of the columns **represent technical specification of the ride.** The definition of all the columns it is presented in the following sections. The data set also **contains the values of the three metrics: excitement, nausea and fear** that we collapsed in one binary flag.

The other data set cane be found online at link and consists of **real world rides present in the US**. This data set in addition to technical specs of the rides also contains information about the year a ride opened in and where the ride is around the country, that is the geographical position. With this information it is possible to **analyze how rides evolved during time**.

The third data set will be created during the **preprocessing phase**.

1. Importing Tools

Importing all the needed packages. Basically we need libraries to preprocess data and then making some visualizations.

```
# renv lockdown file present in the project directory --> renv.lock
knitr::opts_chunk$set(warning = FALSE, message = FALSE, tidy = TRUE)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

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

```
library(forcats)
library(GGally) #ggplot2 extensions, more details in the following sections
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(sf)
```

```
## Linking to GEOS 3.9.1, GDAL 3.2.3, PROJ 7.2.1; sf_use_s2() is TRUE
```

```
sf::sf_use_s2(FALSE)
```

```
## Spherical geometry (s2) switched off
```

```
library(rnaturalearth)
```

2. Data Reading

Reading and importing the **first data set**.

```
game_data<-read_csv("./Data/PlanetCoaster_DF.csv", col_names=TRUE)
head(game_data)
```

```
## # A tibble: 6 x 23
##   Exitement  Fear Nausea Duration Track_Lenght Traversal_Lenght Max_Speed
##       <dbl> <dbl>  <dbl>    <dbl>        <dbl>            <dbl>     <dbl>
## 1      5.3   3.65   1.84     87.8          419             1247        98
## 2      6.2   4.56   1.71     45.7          671               NA        97
## 3      6.79  5      2.59     71.4         1411               NA       123
## 4      6.51  4.12   1.53     67.9         1222               NA       110
## 5      6.5   5.16   2.51    143.          1552               NA       126
## 6      5.52  3.61   1.1     105.          1006               NA        97
## # ... with 16 more variables: Avg_Speed <dbl>, Biggest_Drop <dbl>,
## #   Max_Lateral_g <dbl>, Max_Vertical_g <dbl>, Min_Vertical_g <dbl>,
## #   Max_Forward_g <dbl>, Min_Forward_g <dbl>, Inversions_Num <dbl>,
## #   Airtime_Count <dbl>, Tot_Airtime_Duration <dbl>, Coaster_ID <dbl>,
## #   Coaster_Type <chr>, Coaster_Name <chr>, Launched <dbl>, Shuttle <dbl>,
## #   Coaster_Category <chr>
```

Some information about the columns and the type of values can be obtained using the *str()* function. We spot some columns that need a type changement because are categorical values.

```
#print columns
str(game_data)
```

```
## spec_tbl_df [142 x 23] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Exitement           : num [1:142] 5.3 6.2 6.79 6.51 6.5 5.52 6.95 7.91 5.33 6.79 ...
##  $ Fear                : num [1:142] 3.65 4.56 5 4.12 5.16 3.61 5.38 5.39 2.89 5.38 ...
##  $ Nausea              : num [1:142] 1.84 1.71 2.59 1.53 2.51 1.1 1.87 2.1 1.47 2.1 ...
##  $ Duration            : num [1:142] 87.8 45.7 71.4 67.9 143.4 ...
##  $ Track_Lenght        : num [1:142] 419 671 1411 1222 1552 ...
##  $ Traversal_Lenght    : num [1:142] 1247 NA NA NA NA ...
##  $ Max_Speed           : num [1:142] 98 97 123 110 126 97 116 207 103 149 ...
##  $ Avg_Speed           : num [1:142] 51 53 71 65 39 35 59 84 23 55 ...
##  $ Biggest_Drop        : num [1:142] 34 7 35 37 61 36 39 145 36 85 ...
##  $ Max_Lateral_g       : num [1:142] 3.36 4.03 2.67 2.66 5.04 1.69 2.38 1.18 1.82 3.18 ...
##  $ Max_Vertical_g      : num [1:142] 4.67 4.3 6.37 5.79 6.85 5.7 5.59 6.34 7.95 6.59 ...
```

```
##  $ Min_Vertical_g      : num [1:142] -1.17 -0.42 -0.7 -2.38 -1.53 -0.3 -2.32 -1.32 -1.48 -0.58 ...
##  $ Max_Forward_g       : num [1:142] 1.05 0.82 1.05 0.53 0.71 0.59 2.1 1.71 0.87 0.71 ...
##  $ Min_Forward_g       : num [1:142] -0.74 -1.46 -1.05 -0.84 -1.06 -1.14 -0.73 -1.55 -0.62 -0.91 ...
##  $ Inversions_Num      : num [1:142] 8 0 6 8 0 4 9 0 0 13 ...
##  $ Airtime_Count       : num [1:142] 0 3 2 2 11 0 1 2 7 2 ...
##  $ Tot_Airtime_Duration: num [1:142] 0 2.4 1.4 0.6 8.5 0 0.5 4.2 2.4 1.5 ...
##  $ Coaster_ID          : num [1:142] 1 2 3 4 5 6 7 8 9 10 ...
##  $ Coaster_Type        : chr [1:142] "Impulse" "Motorbike" "Wing" "Wing" ...
##  $ Coaster_Name        : chr [1:142] "Reversed Loop" "BikeRide" "Raptor Runner" "Afterburn" ...
##  $ Launched            : num [1:142] 1 1 1 1 0 0 1 1 0 0 ...
##  $ Shuttle             : num [1:142] 1 0 0 0 0 0 0 0 0 0 ...
##  $ Coaster_Category    : chr [1:142] "Steel" "Steel" "Steel" "Steel" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Exitement = col_double(),
##   ..   Fear = col_double(),
##   ..   Nausea = col_double(),
##   ..   Duration = col_double(),
##   ..   Track_Lenght = col_double(),
##   ..   Traversal_Lenght = col_double(),
##   ..   Max_Speed = col_double(),
##   ..   Avg_Speed = col_double(),
##   ..   Biggest_Drop = col_double(),
##   ..   Max_Lateral_g = col_double(),
##   ..   Max_Vertical_g = col_double(),
##   ..   Min_Vertical_g = col_double(),
##   ..   Max_Forward_g = col_double(),
##   ..   Min_Forward_g = col_double(),
##   ..   Inversions_Num = col_double(),
##   ..   Airtime_Count = col_double(),
##   ..   Tot_Airtime_Duration = col_double(),
##   ..   Coaster_ID = col_double(),
##   ..   Coaster_Type = col_character(),
##   ..   Coaster_Name = col_character(),
##   ..   Launched = col_double(),
##   ..   Shuttle = col_double(),
##   ..   Coaster_Category = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

2.1 Game Ride Data Set Variables Description

The data set contains **23 columns**, but not all are important for the analysis. Here a brief description of the columns:

- **Excitement**: The level of excitement of the ride, numerical value.
- **Fear**: The level of fear of the ride, numerical value.
- **Nausea**: The level of nausea of the ride, numerical value.
- **Duration**: The duration of the ride in seconds.
- **Track_Length**: The length of the track in meters.
- **Traversal_Length**: This apply only to few observations and refers to the "geographical space" occupied by the ride. It is not important in our analysis.
- **Max_Speed**: The maximum speed reached by the train in km/h.
- **Avg_Speed**: The average speed of the train in km/h.
- **Biggest_Drop**: Length of largest gap between high and low points of roller coaster in meters.

- **Max_Lateral_g**: Maximum lateral g force perceived by riders, expressed in g-force.
- **Max_Vertical_g**: Maximum vertical g force perceived by riders, expressed in g-force.
- **Min_Vertical_g**: Minimum lateral g force perceived by riders, expressed in g-force.
- **Max_Forward_g**: Maximum forward g force perceived by riders, expressed in g-force.
- **Min_Forward_g**: Minimum forward g force perceived by riders, expressed in g-force.
- **Inversions_Num**: Number of times roller coaster flips passengers, numerical value.
- **Airtime_Count**: Number of all airtime perceived during the ride, numerical value. Airtime means when riders of a roller coaster experience either weightlessness or negative G-forces.
- **Tot_Airtime_Duration**: Total time of airtime expressed in seconds.
- **Coaster_ID**: Id of the coaster, incremental numerical value.
- **Coaster_Type**: How a passenger is positioned in the roller coaster. There are 34 different roller coaster types in the data set. This type refers to the style of the track and the style of the train, how passengers are seated or not seated and also the kind of restrictions.
- **Coaster_Name**: Name of the ride, string.
- **Launched**: Flag indicating if the ride uses a launching system to propel the train at a speed. 0 if not launched, 1 if launched.
- **Shuttle**: Flag indicating if the ride is shuttle or not. 0 if not shuttle, 1 if shuttle. A shuttle ride is not a closed circuit ride meaning the train runs also backward to come back to the station.
- **Coaster_Category**: The main category of the coaster –> Steel = ride made of steel, Wooden = ride made of wood, Hybrid = ride made with parts of steel and parts of wood.

Reading and importing the **second data set** about the real rides.

```
real_data<-read_csv("./Data/RollerCoastersMontanaUni.csv", col_names = TRUE)
head(real_data)
```

```
## # A tibble: 6 x 15
##   Age_Group Coaster       Park    City  State Type  Design Year_Opened Top_Speed
##   <chr>     <chr>         <chr>   <chr> <chr> <chr> <chr>        <dbl>     <dbl>
## 1 1:older   Zippin Pippin Libert~ Memp~ Tenn~ Wood~ Sit D~        1915        40
## 2 1:older   Jack Rabbit   Kennyw~ West~ Penn~ Wood~ Sit D~        1921        45
## 3 1:older   Thunderhawk   Dorney~ Alle~ Penn~ Wood~ Sit D~        1923        45
## 4 1:older   Giant Dipper  Santa ~ Sant~ Cali~ Wood~ Sit D~        1924        55
## 5 1:older   Thunderbolt   Kennyw~ West~ Penn~ Wood~ Sit D~        1924        55
## 6 1:older   Wildcat       Lake C~ Bris~ Conn~ Wood~ Sit D~        1927        48
## # ... with 6 more variables: Max_Height <dbl>, Drop <dbl>, Length <dbl>,
## #   Duration <dbl>, Inversions <chr>, Num_of_Inversions <dbl>
```

Same procedure to look at some more info about the data we are dealing with and if something need to be changed.

```
#print columns
str(real_data)
```

```
## spec_tbl_df [156 x 15] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Age_Group        : chr [1:156] "1:older" "1:older" "1:older" "1:older" ...
##  $ Coaster          : chr [1:156] "Zippin Pippin" "Jack Rabbit" "Thunderhawk" "Giant Dipper" ...
##  $ Park             : chr [1:156] "Libertyland" "Kennywood Park" "Dorney Park" "Santa Cruz Beach Boa
##  $ City             : chr [1:156] "Memphis" "West Mifflin" "Allentown" "Santa Cruz" ...
##  $ State            : chr [1:156] "Tennessee" "Pennsylvania" "Pennsylvania" "California" ...
##  $ Type             : chr [1:156] "Wooden" "Wooden" "Wooden" "Wooden" ...
##  $ Design           : chr [1:156] "Sit Down" "Sit Down" "Sit Down" "Sit Down" ...
##  $ Year_Opened      : num [1:156] 1915 1921 1923 1924 1924 ...
##  $ Top_Speed        : num [1:156] 40 45 45 55 55 48 50 55 50 25 ...
##  $ Max_Height       : num [1:156] 70 40 80 70 70 85 55 90 84 37 ...
```

```
##  $ Drop             : num [1:156] 70 70 65 65 95 78 52 89 78 25 ...
##  $ Length           : num [1:156] 2865 2132 2767 2640 2887 ...
##  $ Duration         : num [1:156] 90 96 78 112 101 75 105 120 105 84 ...
##  $ Inversions       : chr [1:156] "N" "N" "N" "N" ...
##  $ Num_of_Inversions: num [1:156] 0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Age_Group = col_character(),
##   ..   Coaster = col_character(),
##   ..   Park = col_character(),
##   ..   City = col_character(),
##   ..   State = col_character(),
##   ..   Type = col_character(),
##   ..   Design = col_character(),
##   ..   Year_Opened = col_double(),
##   ..   Top_Speed = col_double(),
##   ..   Max_Height = col_double(),
##   ..   Drop = col_double(),
##   ..   Length = col_double(),
##   ..   Duration = col_double(),
##   ..   Inversions = col_character(),
##   ..   Num_of_Inversions = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

2.2 Real Rides Data Set Variables Description

The data set contains **15 columns**. Here a brief description of the columns:

- **Duration**: The duration of the ride in seconds.
- **Length**: The length of the track in feet
- **Top_Speed**: The maximum speed reached by the train in mph.
- **Drop**: Length of largest gap between high and low points of roller coaster in feet.
- **Coaster**: Ride name.
- **Park**: Name of the park where the roller coaster is located.
- **City**: City where the roller coaster is located.
- **State**: State where the roller coaster is located.
- **Type**: The main category of the coaster –> Steel = ride made of steel, Wooden = ride made of wood.
- **Design**: How a passenger is positioned in the roller coaster.
- **Opened**: Year of opening of the ride.
- **Max_Height**: Highest point of roller coaster in feet
- **Inversions**: Whether or not roller coaster flips passengers at any point (Yes or No)
- **Num_of_Inversions**: Number of times roller coaster flips passengers, numerical value.
- **Age_Group**: 1:Older (Built between 1900-1979), 2:Recent (1980-1999), 3:Newest (2000-current)

---

3. Data Cleaning and Preprocessing

Data Cleaning is a fundamental step in data science. The goal is to *clean* the data, that means **fix incorrect, inaccurate, incomplete and missing parts**.

A typical cleaning approach involves this steps:

- Dropping inconsistent/unnecessary columns
- Handle missing data

- Handle outliers
- Tidy the dataset
- Check data types

3.1 Tidy Dataset

By definition, a data set is tidy when:

1. **Each variable is a column**
2. **Each observation is a row**
3. **Each type of observational unit is a table**

This two data sets are tidy. Also the third one we will create it's tidy.

3.2 Remove Useless Columns

For the game ride data set we select all the columns except for Coaster_Name and Traversal_Length. The names of the rides and the traversal lengths are not important in our analysis.

```
# remove useless columns
game_data <- select(game_data, -Coaster_Name, -Traversal_Lenght)
```

For the real ride data set we select all the columns except for Coaster that refers to the ride name and Inversions. The number of inversion is sufficient for understanding if a coaster has or not at last one inversion.

```
# remove useless columns
real_data <- select(real_data, -Coaster, -Inversions)
```

3.3 Handle Missing Data

With the following instruction we check the presence of null values. In the real rides data set there are **some null value to filter out**.

```
# count the number of nulls
cat("NA in game dataset: ", sum(is.na(game_data)),
    " NA in real dataset: ", sum(is.na(real_data)))
```

```
## NA in game dataset:  0  NA in real dataset:  10
```

We can proceed filtering the data set in order to **remove all the missing values**. We can check again the presence of nulls with the instruction above to ensure we have 0.

```
# dropping NA
real_data <- real_data %>%
  drop_na()
```

3.4 Converting Units

Some of the columns in the real ride data set have different measurement units. We have to **convert mph to km/h and feet to meters**.

```
# converting units
real_data["Top_Speed"]=round(real_data$Top_Speed*1.60, digits = 0) #mph to kmh
real_data["Drop"]=round(real_data$Drop/3.28, digits = 0) #ft to meters
real_data["Length"]=round(real_data$Length/3.28, digits = 0) #ft to meters
real_data["Max_Height"]=round(real_data$Max_Height/3.28, digits = 0) #ft to meters
```

3.5 Check Types

Some columns of the data set are not in the **correct type**, for example some coaster categories. Also the number of inversions is always an integer. Let's fix it.

```
# changing types to factor or integer
game_data$Inversions_Num=as.integer(game_data$Inversions_Num)
game_data$Coaster_Type=as.factor(game_data$Coaster_Type)
game_data$Launched=as.factor(game_data$Launched)
game_data$Shuttle=as.factor(game_data$Shuttle)
game_data$Coaster_Category=as.factor(game_data$Coaster_Category)
```

Check the new types. Even if some attributes are integer in this data set (for example speed, drop, ...), i am not changing these to integer, because these measures can actually be double and maybe in the future we will need to store them as double too.

```
# new data types
str(game_data)
```

```
## tibble [142 x 21] (S3: tbl_df/tbl/data.frame)
##  $ Exitement           : num [1:142] 5.3 6.2 6.79 6.51 6.5 5.52 6.95 7.91 5.33 6.79 ...
##  $ Fear                : num [1:142] 3.65 4.56 5 4.12 5.16 3.61 5.38 5.39 2.89 5.38 ...
##  $ Nausea              : num [1:142] 1.84 1.71 2.59 1.53 2.51 1.1 1.87 2.1 1.47 2.1 ...
##  $ Duration            : num [1:142] 87.8 45.7 71.4 67.9 143.4 ...
##  $ Track_Lenght        : num [1:142] 419 671 1411 1222 1552 ...
##  $ Max_Speed           : num [1:142] 98 97 123 110 126 97 116 207 103 149 ...
##  $ Avg_Speed           : num [1:142] 51 53 71 65 39 35 59 84 23 55 ...
##  $ Biggest_Drop        : num [1:142] 34 7 35 37 61 36 39 145 36 85 ...
##  $ Max_Lateral_g       : num [1:142] 3.36 4.03 2.67 2.66 5.04 1.69 2.38 1.18 1.82 3.18 ...
##  $ Max_Vertical_g      : num [1:142] 4.67 4.3 6.37 5.79 6.85 5.7 5.59 6.34 7.95 6.59 ...
##  $ Min_Vertical_g      : num [1:142] -1.17 -0.42 -0.7 -2.38 -1.53 -0.3 -2.32 -1.32 -1.48 -0.58 ...
##  $ Max_Forward_g       : num [1:142] 1.05 0.82 1.05 0.53 0.71 0.59 2.1 1.71 0.87 0.71 ...
##  $ Min_Forward_g       : num [1:142] -0.74 -1.46 -1.05 -0.84 -1.06 -1.14 -0.73 -1.55 -0.62 -0.91 ...
##  $ Inversions_Num      : int [1:142] 8 0 6 8 0 4 9 0 0 13 ...
##  $ Airtime_Count       : num [1:142] 0 3 2 2 11 0 1 2 7 2 ...
##  $ Tot_Airtime_Duration: num [1:142] 0 2.4 1.4 0.6 8.5 0 0.5 4.2 2.4 1.5 ...
##  $ Coaster_ID          : num [1:142] 1 2 3 4 5 6 7 8 9 10 ...
##  $ Coaster_Type        : Factor w/ 34 levels "Bobsled","Boomerang",..: 10 19 31 31 32 12 27 27 29 8
##  $ Launched            : Factor w/ 2 levels "0","1": 2 2 2 2 1 1 2 2 1 1 ...
##  $ Shuttle             : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 1 1 ...
##  $ Coaster_Category    : Factor w/ 3 levels "Hybrid","Steel",..: 2 2 2 2 3 2 2 2 2 1 ...
```

Now we have to do the same for the **real ride data set**.

```
# changing type into factor or integer
real_data$Age_Group=as.factor(real_data$Age_Group)
real_data$Type=as.factor(real_data$Type)
real_data$Design=as.factor(real_data$Design)
real_data$Year_Opened=as.integer(real_data$Year_Opened)
real_data$Num_of_Inversions=as.integer(real_data$Num_of_Inversions)
```

Check the new types.

```
str(real_data)
```

```
## tibble [148 x 13] (S3: tbl_df/tbl/data.frame)
##  $ Age_Group        : Factor w/ 3 levels "1:older","2:recent",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Park             : chr [1:148] "Libertyland" "Kennywood Park" "Dorney Park" "Santa Cruz Beach Boa
```

```
##  $ City             : chr [1:148] "Memphis" "West Mifflin" "Allentown" "Santa Cruz" ...
##  $ State            : chr [1:148] "Tennessee" "Pennsylvania" "Pennsylvania" "California" ...
##  $ Type             : Factor w/ 2 levels "Steel","Wooden": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Design           : Factor w/ 7 levels "Flying","Inverted",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ Year_Opened      : int [1:148] 1915 1921 1923 1924 1924 1927 1935 1940 1946 1951 ...
##  $ Top_Speed        : num [1:148] 64 72 72 88 88 77 80 88 80 40 ...
##  $ Max_Height       : num [1:148] 21 12 24 21 21 26 17 27 26 11 ...
##  $ Drop             : num [1:148] 21 21 20 20 29 24 16 27 24 8 ...
##  $ Length           : num [1:148] 873 650 844 805 880 ...
##  $ Duration         : num [1:148] 90 96 78 112 101 75 105 120 105 84 ...
##  $ Num_of_Inversions: int [1:148] 0 0 0 0 0 0 0 0 0 0 ...
```

3.6 Adding a class to each ride - Good vs Bad coasters

Another important step for our analysis is create a **flag in the game rides that "summarize" the three evaluation metrics of a ride**. This flag, called *coaster_class* indicates if a rides it's Good or Bad, based on the definition of a good ride we gave in the initial description.

```
#add coaster class as factor
game_data <- game_data %>%
  mutate(coaster_class = as.factor(if_else
        (Exitement>6.34 & Nausea<3.5 & (Fear>4.0 & Fear<6.0), "Good", "Bad")))
```

3.7 Creating the third data set with real and game rides together

In order to compare the real and game rides, we need to create a new data set that contains **all the observation from the game and real data** set. To do so, we first **select the attributes** in common, and then we **give the same names** to all the columns.

```
# select the common attributes of real and game rides
game <-
select(game_data, Max_Speed, Biggest_Drop, Track_Lenght, Duration, Inversions_Num, -coaster_class)
real <-
select(real_data, Top_Speed, Drop, Length, Duration, Num_of_Inversions)
# renaming real cols to have same name of game cols
real <-
rename(real, Max_Speed=Top_Speed, Biggest_Drop=Drop, Track_Lenght=Length,
       Inversions_Num=Num_of_Inversions)
```

Then we add a new column that indicates **if the ride is real or game created**. At the end the two data sets are merged together.

```
# adding flag to identify real vs game rides
game["Ride_Type"]="Game"
real["Ride_Type"]="Real"

# merging datasets
rides <- rbind(game, real)

# resulting data set
head(rides)
```

```
## # A tibble: 6 x 6
##   Max_Speed Biggest_Drop Track_Lenght Duration Inversions_Num Ride_Type
##       <dbl>        <dbl>        <dbl>    <dbl>          <int> <chr>
## 1        98           34          419     87.8              8 Game
## 2        97            7          671     45.7              0 Game
```

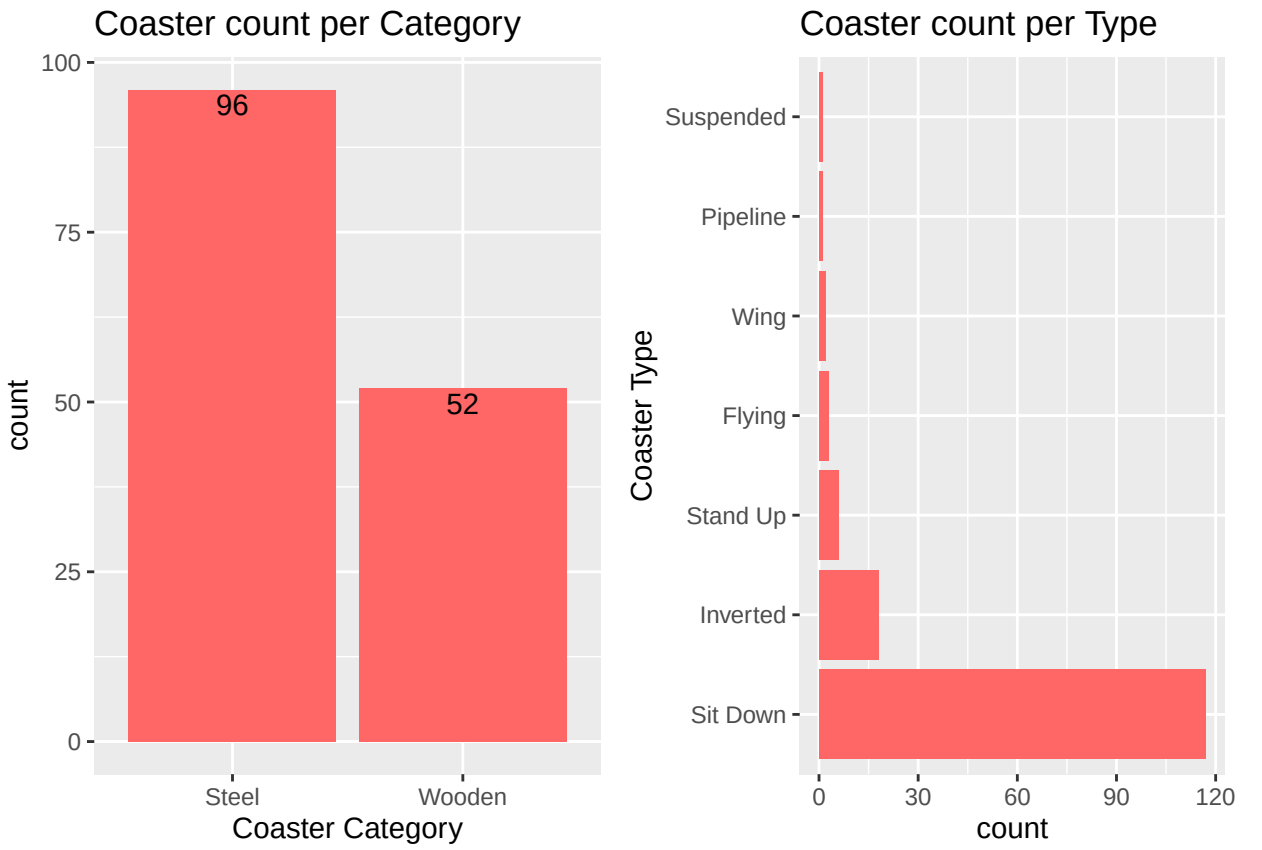
```
## 3       123           35         1411     71.4            6 Game
## 4       110           37         1222     67.9            8 Game
## 5       126           61         1552    143.             0 Game
## 6        97           36         1006    105.             4 Game
```

Now **all the three data set are ready** to be used for the next steps. Let's start with some visualization about the real rides.

4. Real Rides Data Visualization

We start by plotting some information about the number of observations. As we see from the following plots, the **most popular rides are steel coasters**, and regarding to the type, sit down coaster are the majority.

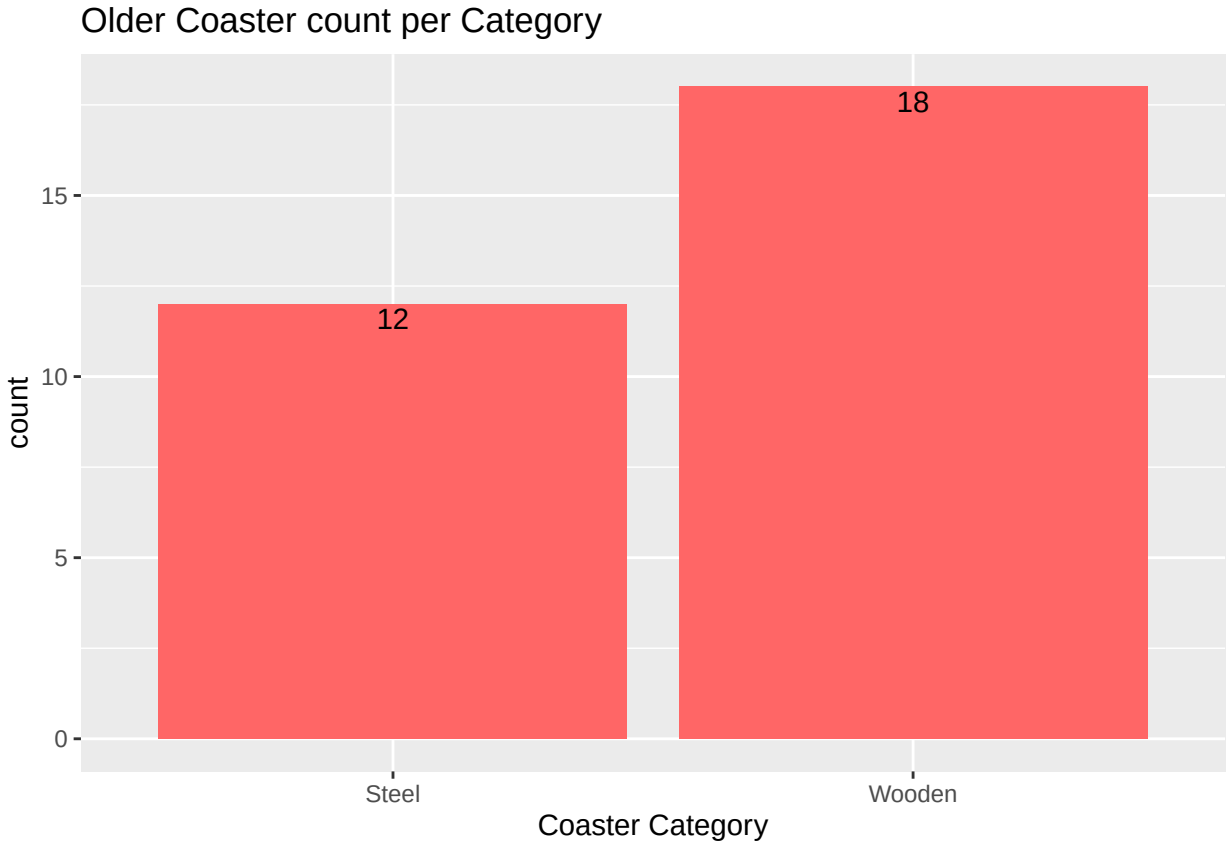
```
# coasters per category (type)
coasterCat <- ggplot(data=real_data, mapping = aes(x = Type))+
  geom_bar(fill="#FF6666")+
  geom_text(mapping = aes(label=..count..), stat = "count", vjust=1.2)+
  labs(title = "Coaster count per Category", x = "Coaster Category")
# coaster per type (design)
coasterDes <- ggplot(data=real_data, mapping = aes(x = fct_infreq(Design)))+
  geom_bar(fill="#FF6666")+
  coord_flip()+
  labs(title = "Coaster count per Type", x = "Coaster Type")

# arranging the plots
grid.arrange(coasterCat, coasterDes, nrow=1)
```



Given that steel coasters are the most popular, let's see if this is true also for the oldest rides. The following

plot show that this is not true, in fact, **in the past the most common coaster belonged to the wooden category** this maybe because wooden was less expensive in the past and also because wooden coaster are built in a manner that allowed not to use large cranes and advanced building systems, not available in the past. Now this tools are used to speed up the process.

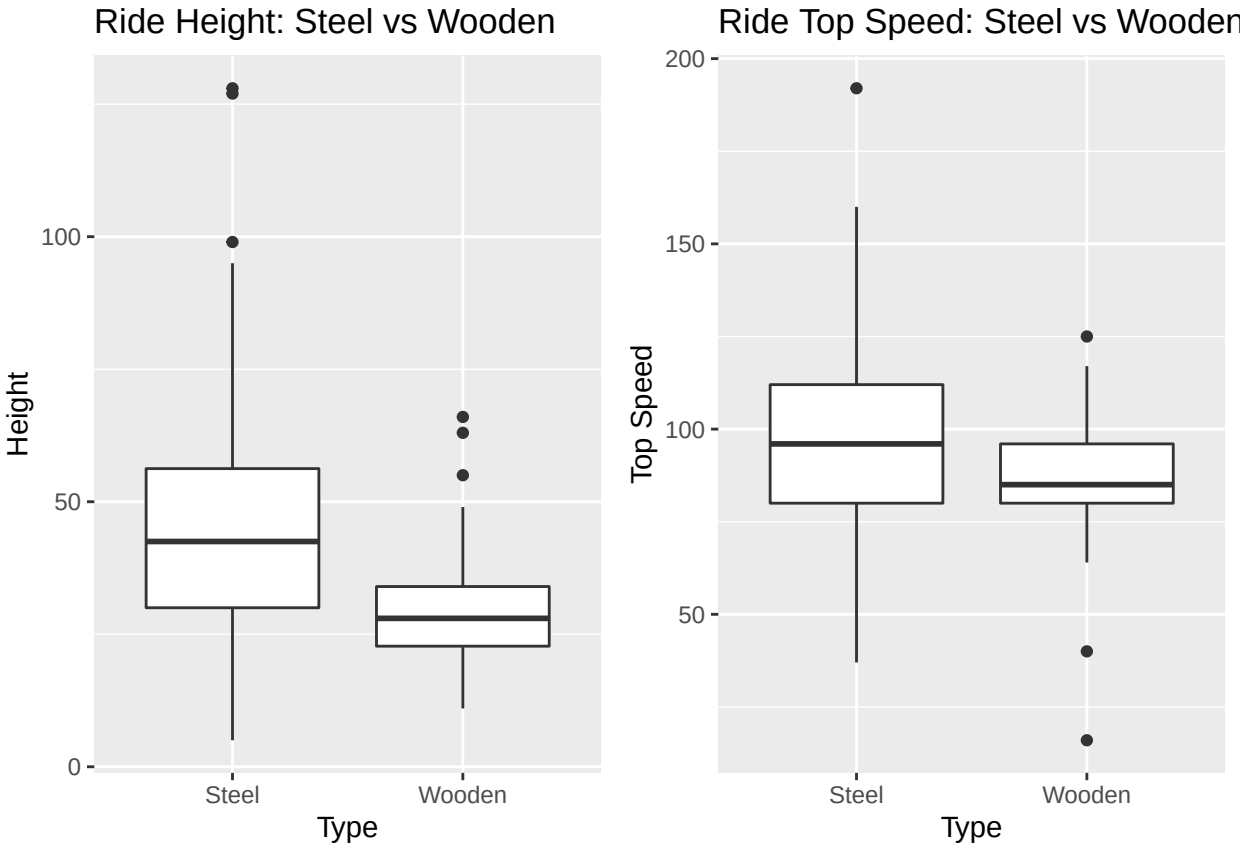
```
# selecting the older rides filtering the age group
real_data %>%
  filter(Age_Group=="1:older")%>%
  ggplot(mapping = aes(x = Type))+
  geom_bar(fill="#FF6666")+
  geom_text(mapping = aes(label=..count..), stat = "count", vjust=1.2)+
  labs(title = "Older Coaster count per Category", x = "Coaster Category")
```



Now it is interesting to understand if wooden coaster can "compete" with the most modern steel coasters in terms of height and speed. The following plots show that **steel coasters are higher and faster** than wooden competitors. The strength of the steel structure allows for fewer supports and greater stability, allowing for more extreme rides.

```
# box plots height vs type
heightComp <- real_data %>%
  ggplot(aes(x=Type, y=Max_Height))+
  geom_boxplot()+
  labs(title = "Ride Height: Steel vs Wooden", y="Height")

# box plots speed vs type
speedComp <- real_data %>%
  ggplot(aes(x=Type, y=Top_Speed))+
  geom_boxplot()+
```

```
  labs(title = "Ride Top Speed: Steel vs Wooden", y="Top Speed")

grid.arrange(heightComp, speedComp, nrow=1)
```

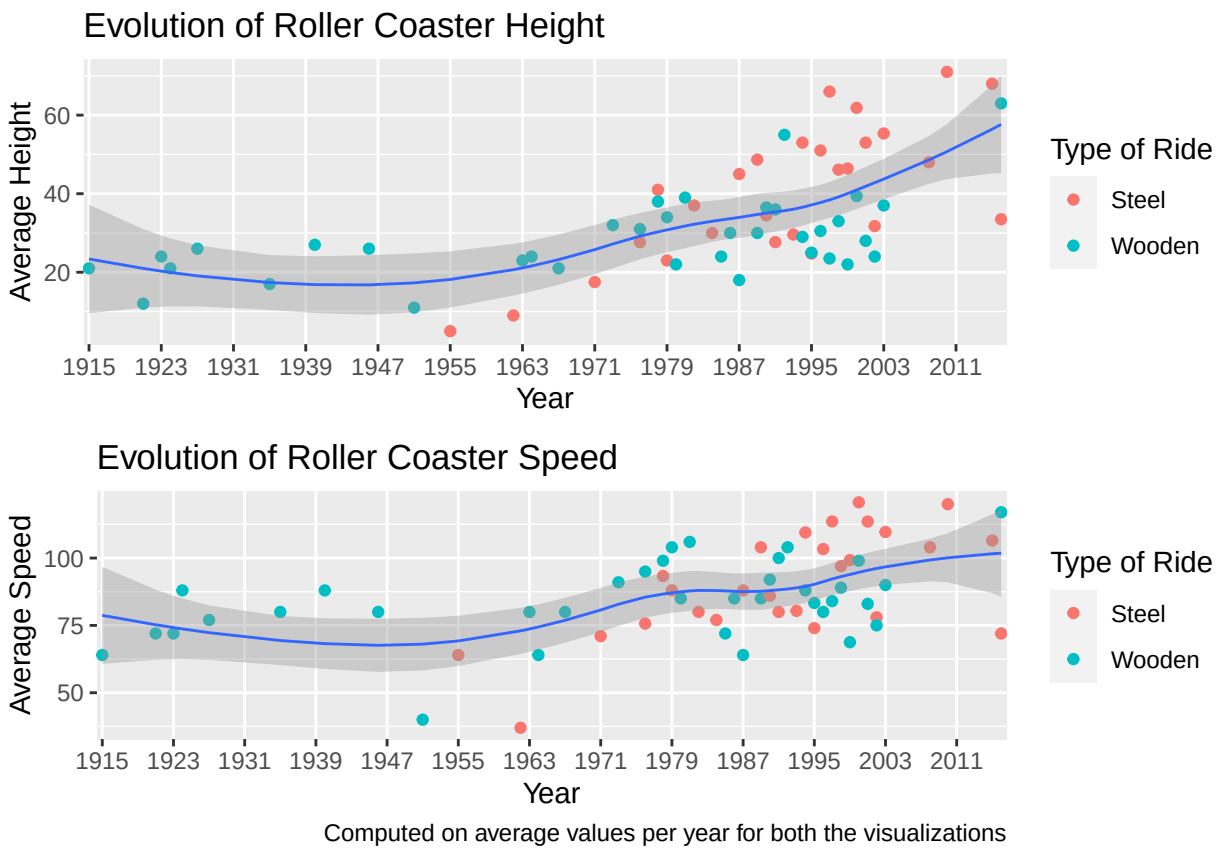


Now we know that steel coasters are the most popular, higher and faster. Wooden coasters used to be the most popular, and even the newest one are slower and lower compared to steel rides.

Let's take now a look at **how speed and height of ride evolved during the years**. Both wooden and steel coaster have experienced growth in terms of height, especially in recent years. The growth in speed is less noticeable. It also interesting to note that the curves are similar, maybe indicating a **relation between speed and height**: if a ride is higher generally the train reaches an higher speed and vice versa.

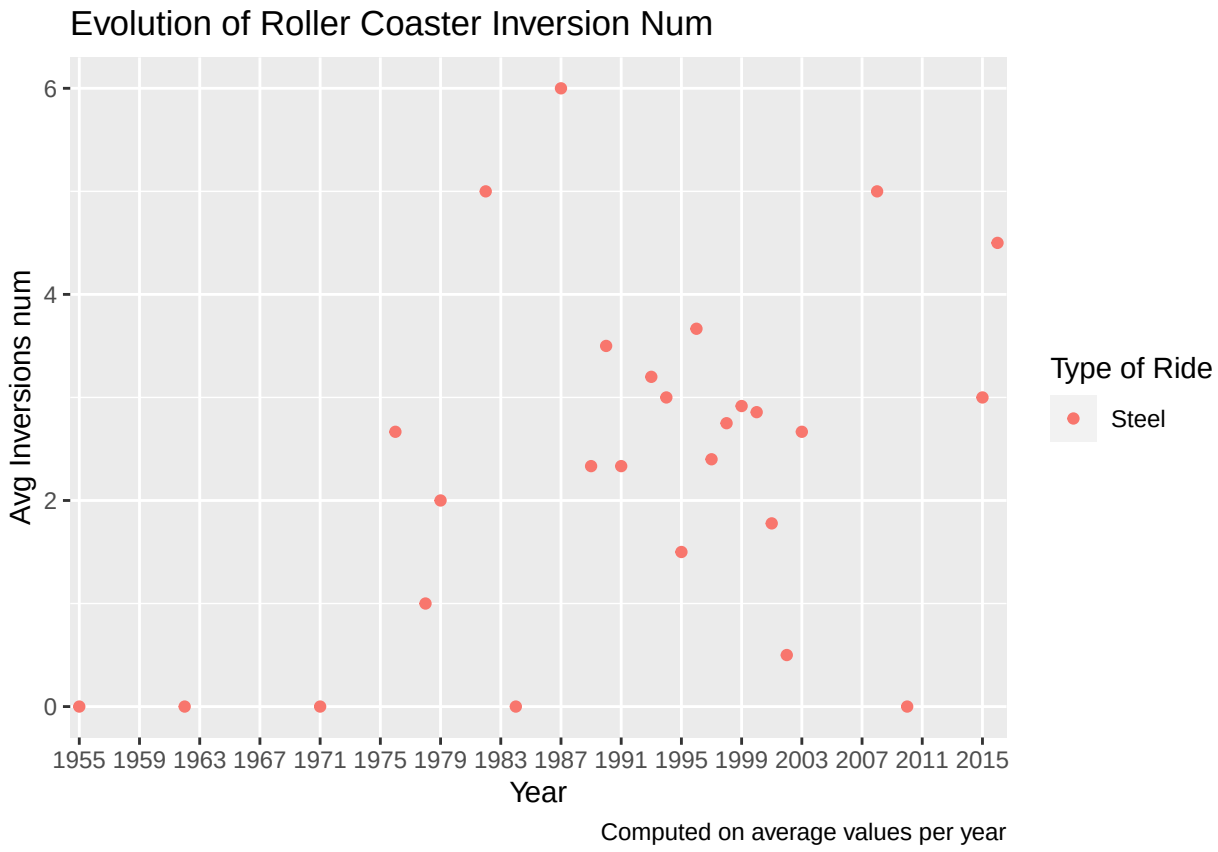
```
# how ride height changes with years
heightEv <- real_data %>%
  group_by(Year_Opened, Type)%>%
  summarise(avgHeightPerYear = mean(Max_Height))%>%
  ggplot()+
  geom_point(aes(x=Year_Opened, y=avgHeightPerYear, color=Type))+
  geom_smooth(aes(x=Year_Opened, y=avgHeightPerYear), size=0.5)+
  scale_x_discrete(limits=seq(1915, 2016, 8))+
  labs(title = "Evolution of Roller Coaster Height", x="Year",
       color="Type of Ride", y="Average Height")


# how ride speed changes with years
speedEv <- real_data %>%
  group_by(Year_Opened, Type)%>%
  summarise(avgSpeedPerYear = mean(Top_Speed))%>%
```

```
  ggplot()+
  geom_point(aes(x=Year_Opened, y=avgSpeedPerYear, color=Type))+
  geom_smooth(aes(x=Year_Opened, y=avgSpeedPerYear), size=0.5)+
    scale_x_discrete(limits=seq(1915, 2016, 8))+
  labs(title = "Evolution of Roller Coaster Speed", x="Year",
       color="Type of Ride", y="Average Speed",
       caption = "Computed on average values per year for both the visualizations")

grid.arrange(heightEv, speedEv, nrow=2)
```



I also believe that during the years the number of inversions changed. First note that Wooden coasters do not have inversions, they cannot invert. So we look at the inversions evolution during years for only the Steel category. We clearly see **after the 1975 a growth in the number of inversions.** This growth seems to have stopped and reached a level of stability.

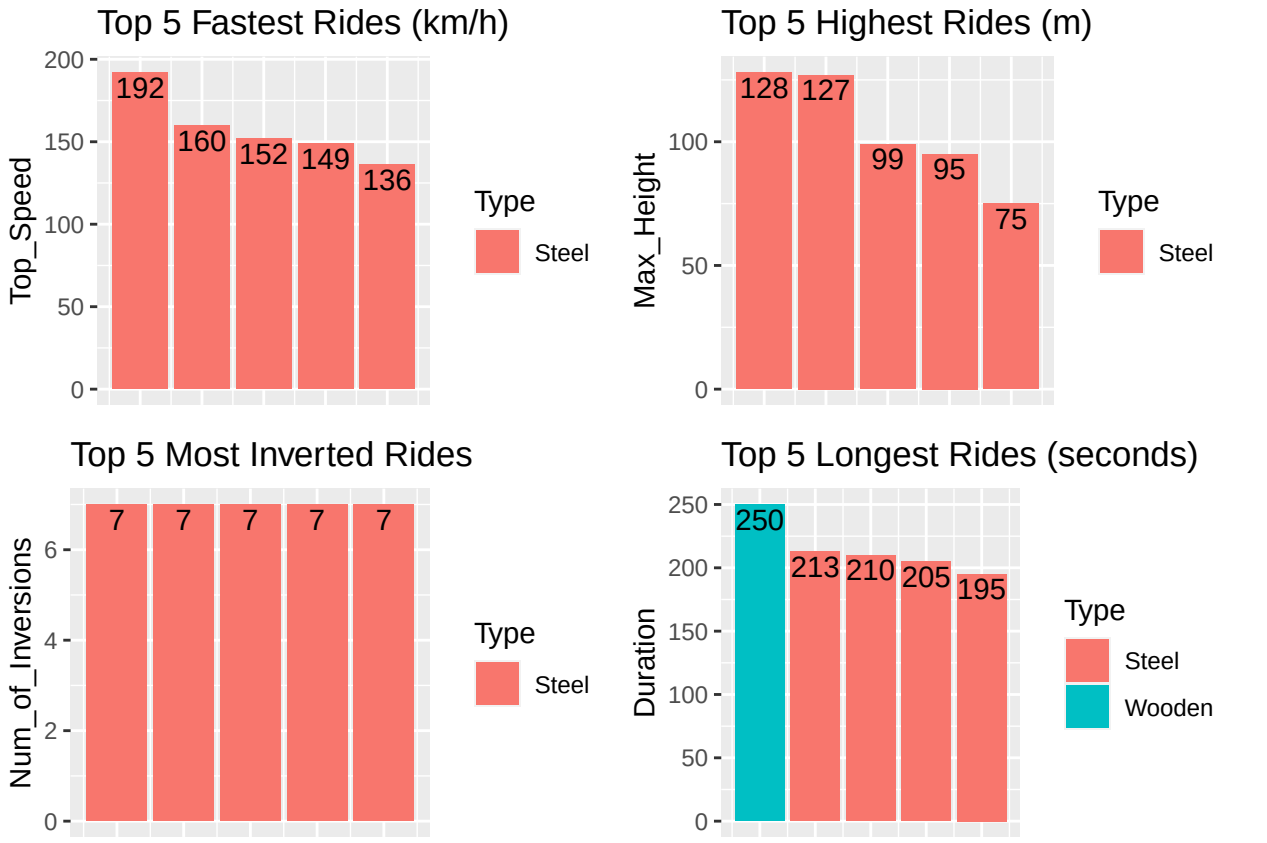
```
# how ride inversions changes with years
real_data %>%
  group_by(Year_Opened, Type)%>%
  filter(Type == "Steel")%>%
  summarise(avgInvsPerYear = mean(Num_of_Inversions))%>%
  ggplot()+
  geom_point(aes(x=Year_Opened, y=avgInvsPerYear, color=Type))+
  scale_x_discrete(limits=seq(1955, 2016, 4))+
  labs(title = "Evolution of Roller Coaster Inversion Num", x="Year",
       color="Type of Ride",
       y="Avg Inversions num", caption = "Computed on average values per year")
```



We proceed by plotting the 5 **fastest, higher, longest and most inverted rides**. This ranks show that rides have a max speed of 192 km/h, a max height of 128 meters and a max of 7 inversions. The longest ride in terms of time lasts 250 seconds, that is 4 minutes. The 5 faster, higher and most inverted rides are only Steel coasters. The longest in terms of seconds is however a Wooden ride.

```
# fastest and highest rides in the us
fastest <- real_data%>%
  slice_max(Top_Speed, n=5, with_ties = FALSE)%>%
  ggplot(aes(x=seq(1,5), y=Top_Speed, fill=Type))+
  geom_bar(stat = "identity")+
  geom_text(mapping = aes(label=Top_Speed), vjust=1.2)+
  labs(title = "Top 5 Fastest Rides (km/h)")+
  theme(axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank())

highest <- real_data%>%
  slice_max(Max_Height, n=5, with_ties = FALSE)%>%
  ggplot(aes(x=seq(1,5), y=Max_Height, fill=Type))+
  geom_bar(stat = "identity")+
  geom_text(mapping = aes(label=Max_Height), vjust=1.2)+
  labs(title = "Top 5 Highest Rides (m)")+
  theme(axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank())

mostInv <- real_data%>%
```

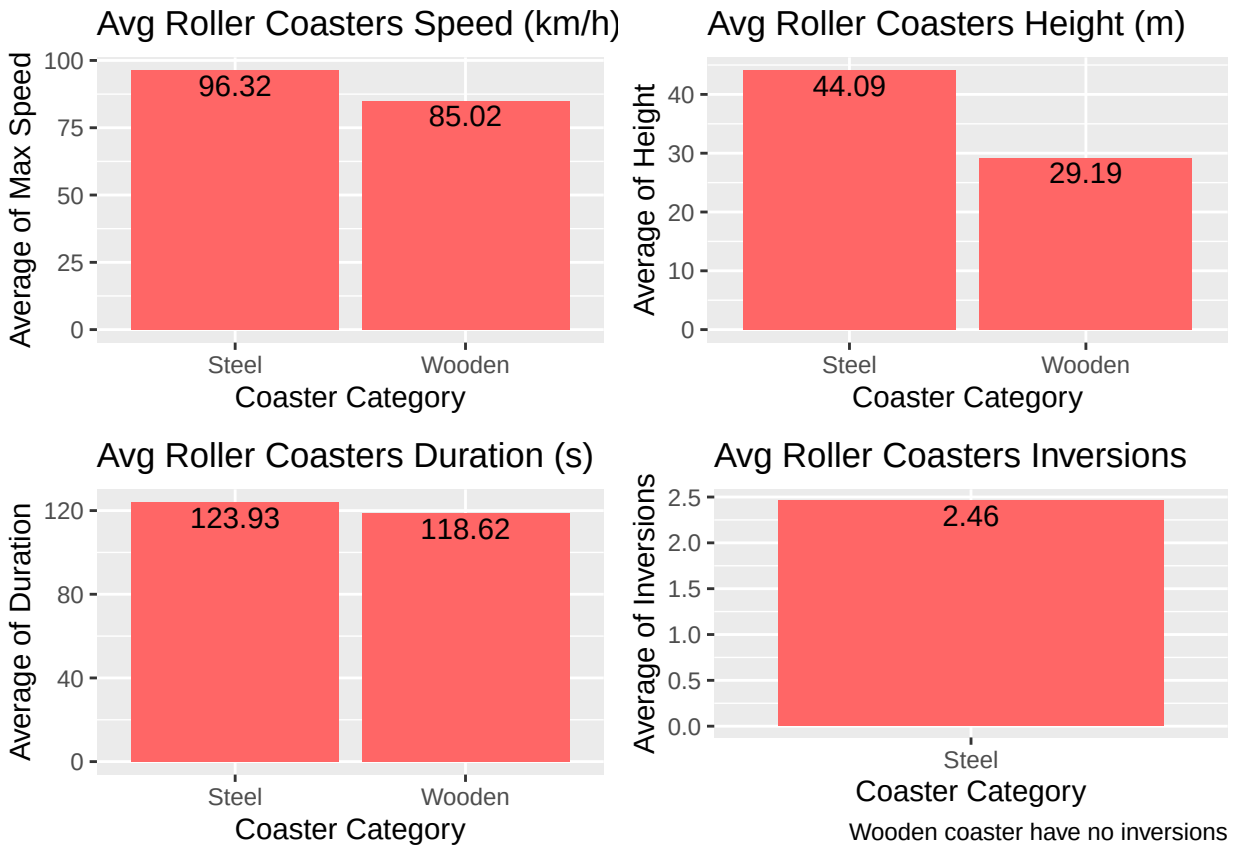
```
  slice_max(Num_of_Inversions, n=5, with_ties = FALSE)%>%
  ggplot(aes(x=seq(1,5), y=Num_of_Inversions, fill=Type))+
  geom_bar(stat = "identity")+
  geom_text(mapping = aes(label=Num_of_Inversions), vjust=1.2)+
  labs(title = "Top 5 Most Inverted Rides")+
  theme(axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank())

longest <- real_data%>%
  slice_max(Duration, n=5, with_ties = FALSE)%>%
  ggplot(aes(x=seq(1,5), y=Duration, fill=Type))+
  geom_bar(stat = "identity")+
  geom_text(mapping = aes(label=Duration), vjust=1.2)+
  labs(title = "Top 5 Longest Rides (seconds)")+
  theme(axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank())

grid.arrange(fastest, highest, mostInv, longest, nrow=2)
```



This are the top 5 values, let's look now at the **average values to see how much differs**. The average values are divided on the category of the ride: wooden or steel. We notice a **difference between the maximum values and the average values**: in real life the are only few extreme coasters, the majority are less extreme. Also in the coaster industry company sells pre-made rides, the are sold all over the world. This **pre-made rides are usually not-so-extreme** and can fit in almost all theme parks. Extreme rides needs to be **custom-built and are more expensive, more difficult to fit and so less popular**. This facts

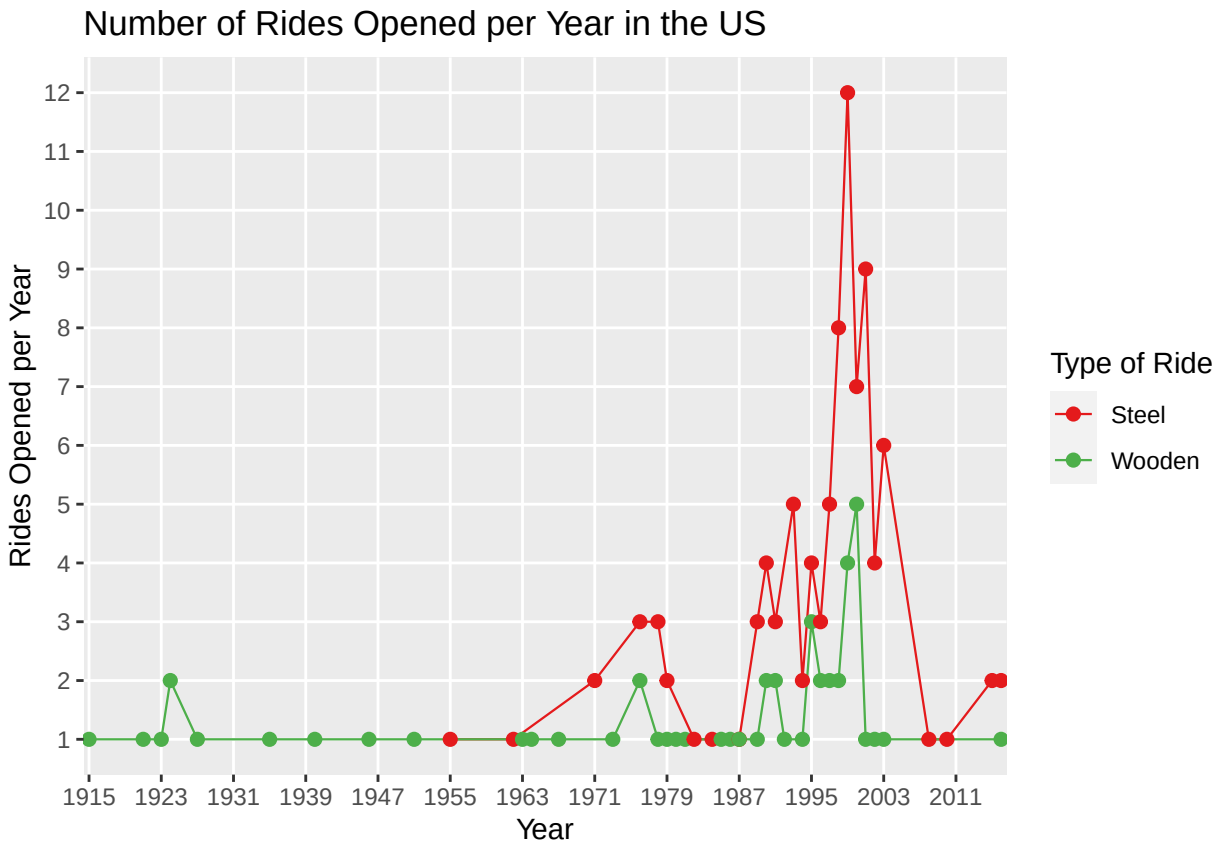
explains the differences between the top rides technical specs and the average ones.

```
# avg speed for coaster cat
avgSpd <- real_data %>%
  group_by(Type)%>%
  summarise(avgSpeed=round(mean(Top_Speed), digits = 2))%>%
  ggplot(aes(x=Type, y=avgSpeed))+
    geom_col(fill="#FF6666")+
    geom_text(mapping = aes(label=avgSpeed), vjust=1.2)+
    labs(title = "Avg Roller Coasters Speed (km/h)",
         x = "Coaster Category", y="Average of Max Speed")

# avg height for coaster cat
avgHeight <- real_data %>%
  group_by(Type)%>%
  summarise(avgHeight=round(mean(Max_Height), digits = 2))%>%
  ggplot(aes(x=Type, y=avgHeight))+
    geom_col(fill="#FF6666")+
    geom_text(mapping = aes(label=avgHeight), vjust=1.2)+
    labs(title = "Avg Roller Coasters Height (m)",
         x = "Coaster Category", y="Average of Height")

# avg duration for coaster cat
avgDur <- real_data %>%
  group_by(Type)%>%
  summarise(avgDura=round(mean(Duration), digits = 2))%>%
  ggplot(aes(x=Type, y=avgDura))+
    geom_col(fill="#FF6666")+
    geom_text(mapping = aes(label=avgDura), vjust=1.2)+
    labs(title = "Avg Roller Coasters Duration (s)",
         x = "Coaster Category", y="Average of Duration")

# avg inversion num for coaster cat
avgInv <- real_data %>%
  group_by(Type)%>%
  filter(Type == "Steel")%>%
  summarise(avgInve=round(mean(Num_of_Inversions), digits = 2))%>%
  ggplot(aes(x=Type, y=avgInve))+
    geom_col(fill="#FF6666")+
    geom_text(mapping = aes(label=avgInve), vjust=1.2)+
    labs(title = "Avg Roller Coasters Inversions",
         x = "Coaster Category", y="Average of Inversions",
         caption = "Wooden coaster have no inversions")

grid.arrange(avgSpd, avgHeight, avgDur, avgInv, nrow = 2)
```



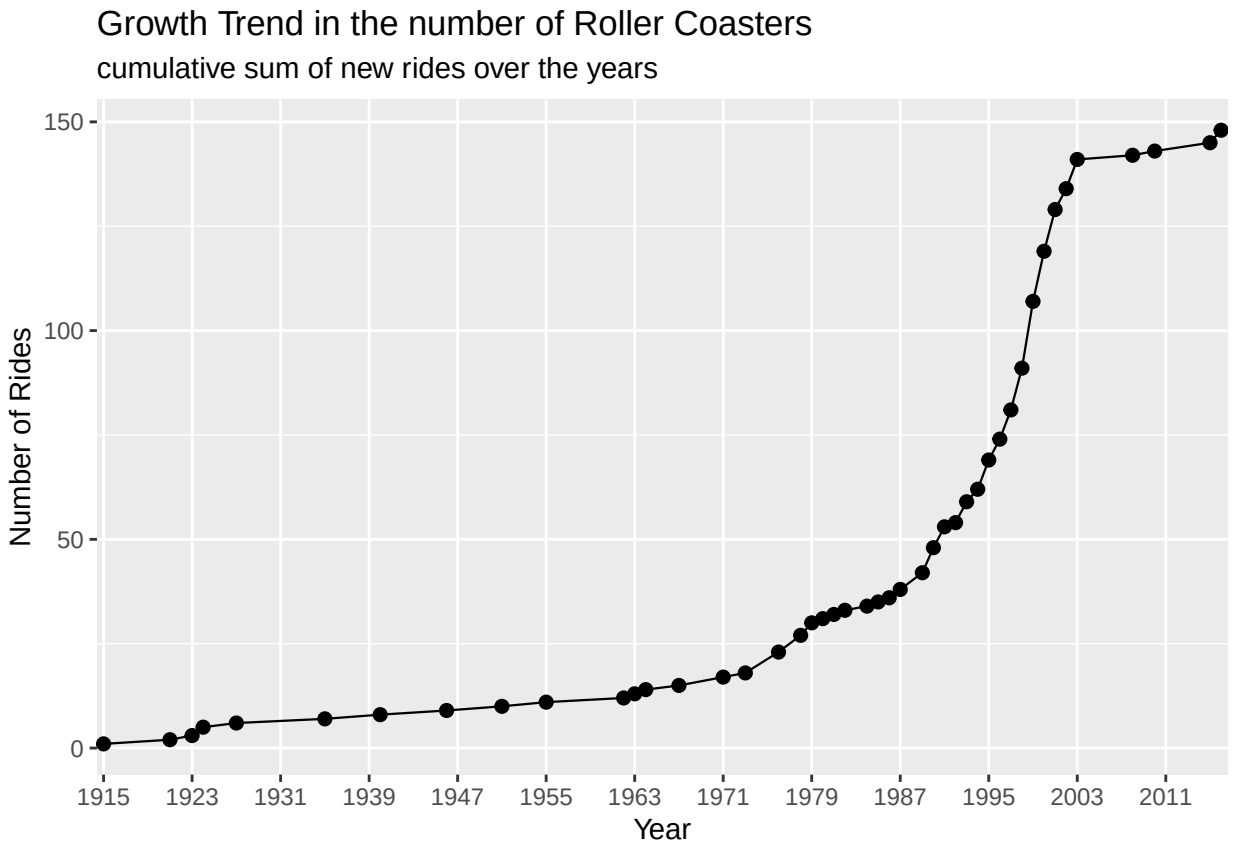
Now we know that roller coaster changed during the years and also how them changed. At this point it is interesting to look at the **evolution of the roller coasters industry** in the US by looking at the number of **rides opened per year**, during different years. The result shows a peak of new rides from the 1990' to the 2000'. The majority of this rides are steel coasters: nowadays steel coasters are faster and easier to build. Wooden coasters anyway are still appreciated.

```
# roller coasters opened per year
real_data %>%
  group_by(Year_Opened, Type)%>%
  summarise(OpenedPerYear = n())%>%
  ggplot(aes(x=Year_Opened, y=OpenedPerYear, color=Type))+
  geom_line(size = 0.4)+
  geom_point(size=2)+
  # limits 1: if i have the year in the dataset, at least one ride was opened
  scale_y_discrete(name="Rides Opened per Year", limits=factor(c(1:12)))+
  scale_x_discrete(limits=seq(1915, 2016, 8))+
  scale_color_manual(values=c('#e41a1c','#4daf4a'))+
  labs(title = "Number of Rides Opened per Year in the US",
       x="Year", color="Type of Ride")
```



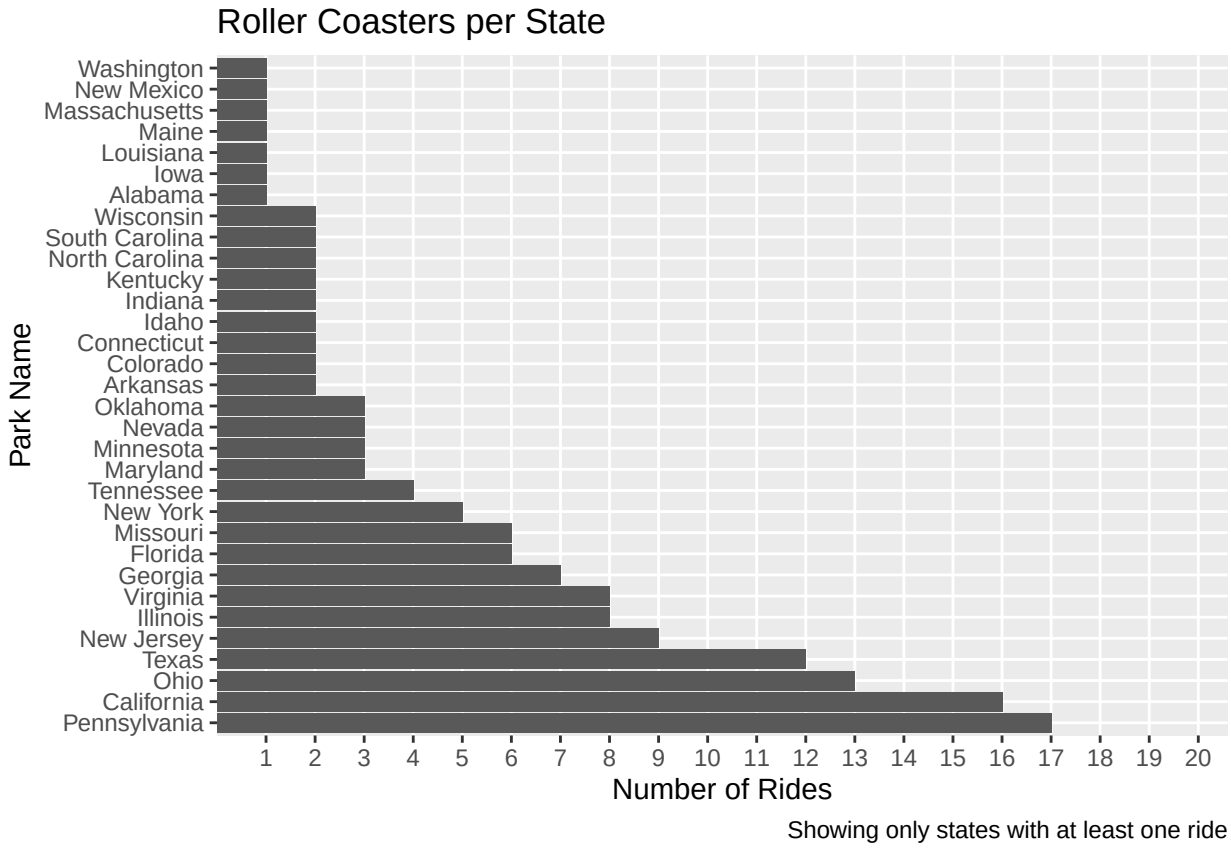
The identified pick is confirmed by the following plot showing the **growth rate in the number of rides during the years**. The steepest the line the fastest the growth. From this plot we also note that growth has **slowed down sharply in recent years**. The reason of this behavior maybe is the fact that new rides costs a lot and once built they remain in the park for a long time, so there is no need to build new ones. Also modern rides require a big space and it's not an easy staff to expand a theme park! **After the big growth in the past years, parks do not have the needs of building a high number of new rides.** At Gardaland for example, italian theme park on Garda lake, some rides are pretty old (Blue Tornado was built in 1998) but are still working.

```
# cumulative sum
real_data %>%
  group_by(Year_Opened)%>%
  summarise(OpenedTot = n())%>%
  mutate(cumSum = cumsum(OpenedTot))%>%
  ggplot(aes(x=Year_Opened, y=cumSum))+
  geom_line(size = 0.4)+
  geom_point(size=2)+
  scale_x_discrete(limits=seq(1915, 2016, 8))+
  scale_color_manual(values=c('#e41a1c','#4daf4a'))+
  labs(title = "Growth Trend in the number of Roller Coasters",
       subtitle = "cumulative sum of new rides over the years",
       x="Year", y="Number of Rides")
```



Now that we have an idea of the evolution of rides during the years, it's time to analyze **how the roller coasters are distributed all over the country**. Let's start with an histogram showing the number of rides present in each state. Some states have a higher number of rides, with a maximum of 17 in Pennsylvania.

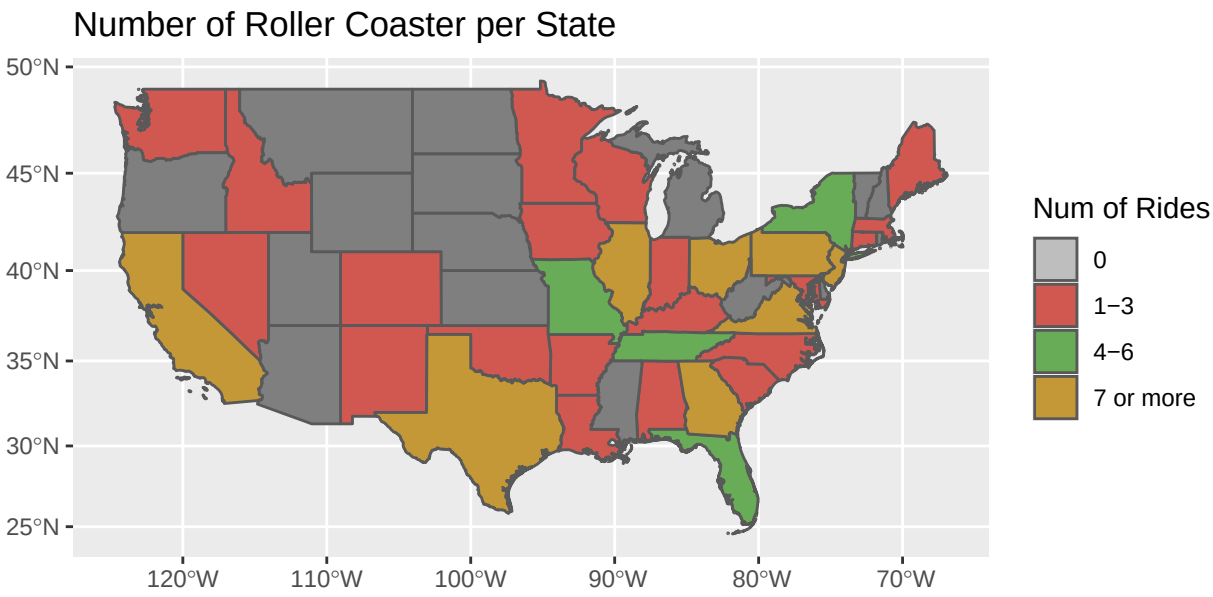
```
# rides num per state
real_data %>%
  group_by(State)%>%
  summarise(nRide = n())%>%
  ggplot(aes(reorder(State, -nRide), nRide))+
  geom_col()+
  coord_flip()+
  scale_y_discrete(name="Number of Rides", limits=factor(c(1:20)))+
  labs(title = "Roller Coasters per State",
       caption = "Showing only states with at least one ride",
       x="Park Name")
```



To have a spatial idea, we can plot the same information using a map. It seems that the states with **the higher number of rides are the ones with higher temperature and a warmer climate**, so that theme parks can be **opened for longer periods, ensuring a better income**.

```
proj <- "+proj=merc"
# get the coasters
coasters <- real_data%>%
  mutate(name = State)%>%
  group_by(name)%>%
  summarise(ridesN=n())%>%
  arrange(name, desc(ridesN))

# get the all states except for Alaska and Hawaii (no rides there)
states <- ne_states(country = "United States of America", returnclass = "sf")%>%
  st_transform(proj)%>%
  group_by(name)%>%
  summarise()%>%
  ungroup()%>%
  left_join(coasters, by="name")%>%
  filter(name != "Alaska", name != "Hawaii")

# plotting
states%>%
  mutate(ridesN = as.factor(ridesN))%>%
  # collapsing only the factors which label is present, avoid warning
  mutate(ridesN = fct_collapse(ridesN,
                               "0" = c(),
```

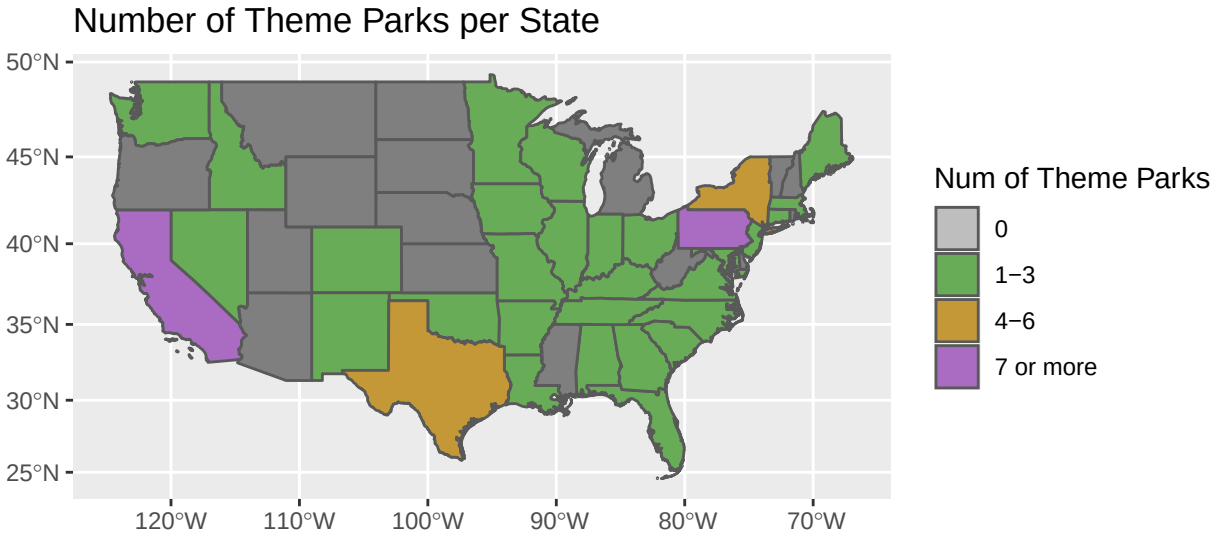
```
                              "1-3" = c("1","2","3"),
                              "4-6" = c("4","5","6"),
                              "7 or more" = c("7","8","9", "12", "13", "16", "17")
                              ))%>%
  ggplot(aes(fill=ridesN))+
  geom_sf()+
  scale_color_manual(values=c("0"="gray", "1-3"="#d05851","4-6"="#69ac57","7 or more"="#c49837"),
                     aesthetics = c('colour', 'fill'))+
  labs(title = "Number of Roller Coaster per State", fill="Num of Rides")
```



As we know, roller coaster are **mainly built within theme parks**. The following map represents the **theme parks around the US**, grouped by the number of rides for each. In this visualization it's even clearer that the states with a majority of theme parks are the one **with a warmer climate**.

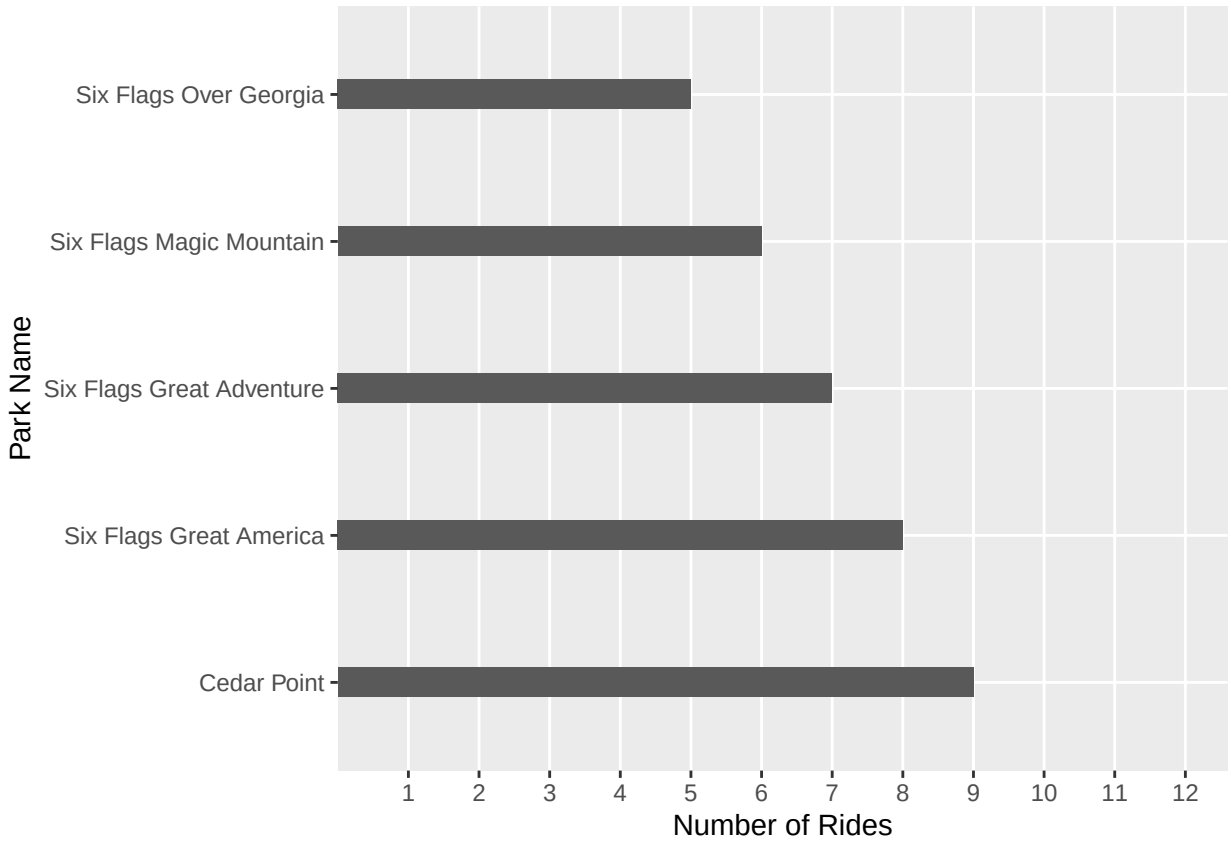
```
proj <- "+proj=merc"
# getting the parks
parks <- real_data%>%
  mutate(name = State)%>%
  group_by(name)%>%
  summarise(parksN=length(unique(Park)))%>%
  arrange(name, desc(parksN))

# get the states except for Alaska and Hawaii (no theme parks there)
states <- ne_states(country = "United States of America", returnclass = "sf")%>%
  st_transform(proj)%>%
  group_by(name)%>%
  summarise()%>%
  ungroup()%>%
  left_join(parks, by="name")%>%
  filter(name != "Alaska", name != "Hawaii")

# plotting
states%>%
  mutate(parksN = as.factor(parksN))%>%
  mutate(parksN = fct_collapse(parksN,
```

```
                              "0" = c(),
                              "1-3" = c("1","2","3"),
                              "4-6" = c("4"),
                              "7 or more" = c("7", "8")
                              ))%>%
  ggplot(aes(fill=parksN))+
  geom_sf()+
  scale_color_manual(values=c("0"="gray","1-3"="#69ac57","4-6"="#c49837","7 or more"="#a96cc2"),
                     aesthetics = c('colour', 'fill'))+
  labs(title = "Number of Theme Parks per State", fill="Num of Theme Parks")
```



I want to conclude this part with a plot representing **the parks with the highest number or roller coasters**. The number one is Cedar Point with 9 rides! (a paradise for coaster enthusiast like me). Another interesting thing is that there are states with even 17 roller coasters, but the biggest park has "only" 9... basically **there are lots of parks around the US with a smaller number of rides instead of a big park with a huge number of coasters**.

```
# parks with 4+ rides
real_data %>%
  group_by(Park)%>%
  summarise(nRide = n())%>%
  filter(nRide>4)%>%
  ggplot(aes(reorder(Park, -nRide), nRide, width=0.2))+
  geom_col()+
  coord_flip()+
  scale_y_discrete(name="Number of Rides", limits=factor(c(1:12)))+
  labs(x="Park Name")
```



4.1 Real Ride Analysis: Conclusions

In this analysis of real rides we understood that **most popular rides changed over years** from wooden coasters to steel coasters, **becoming higher and faster and with more inversions**. Even modern **Wooden coaster cannot compete with Steel coasters**, that remains the choice to build thrilling rides. We understood also that only **few rides are extreme and hold the highest value in terms of technical specs**, the majority are pre-made rides with lower values. Further, the roller coaster industry faced an **intense growth from the 90' to the 2000'** with an high number of new rides. In the US there are states with **lots of different theme parks and rides** and the states with the higher numbers are the ones **with a warmer climate**. The park with the higher number of coasters is Cedar Point, located in Ohio.

5. Game Rides Data Visualization

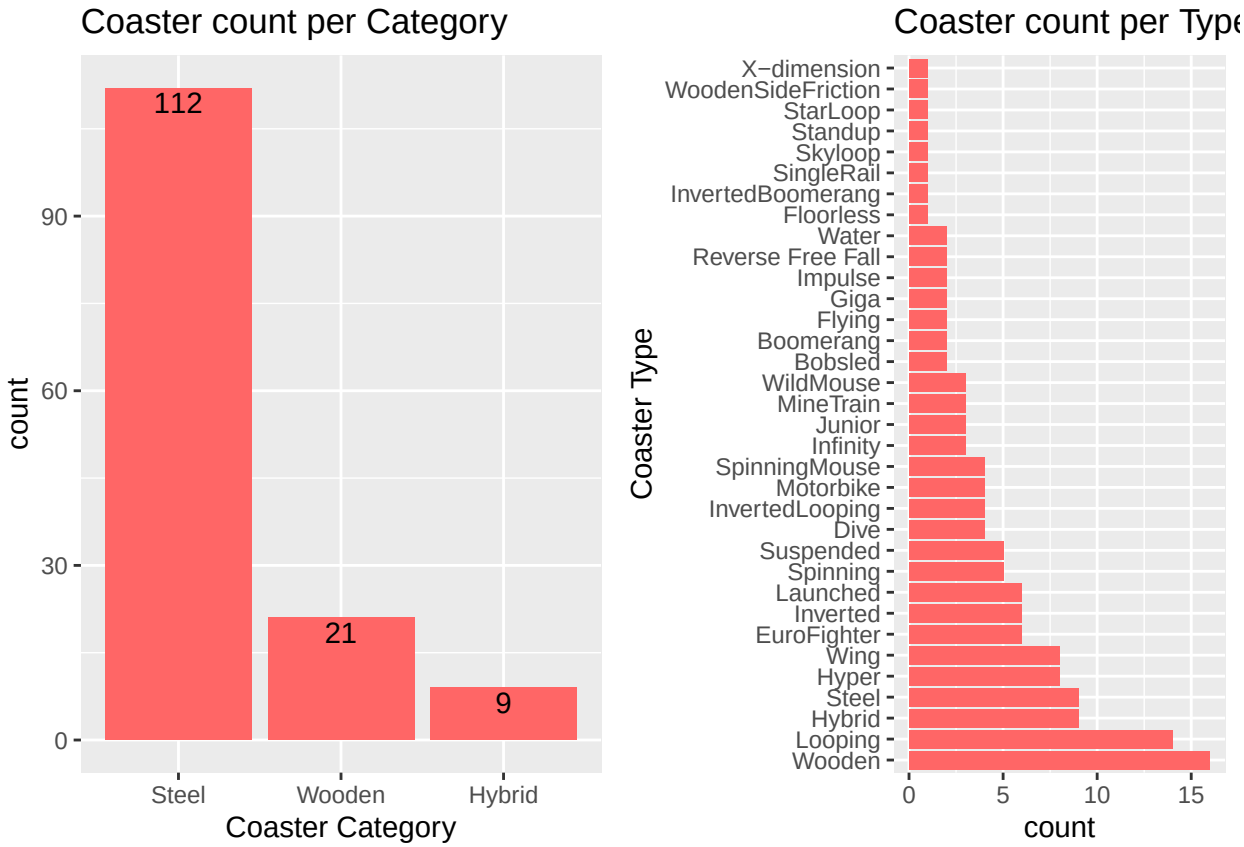
In this section we are going to **analyze the game created rides**: not only the technical specs but also the metric that defines good and bad rides, trying to understand **what affect the rating of a ride**.

Also in this game ride data set the **most popular rides are Steel coasters**, followed by Wooden and Hybrid rides. Talking about coaster types, the most common are Wooden and Looping.

Note that all the types excepted Wooden and Hybrid belong to the Steel category, this explains the high number of steel coasters.

```
coasterMainCat <- ggplot(data=game_data, mapping = aes(x = fct_infreq(Coaster_Category)))+
  geom_bar(fill="#FF6666")+
  geom_text(mapping = aes(label=..count..), stat = "count", vjust=1.2)+
  labs(title = "Coaster count per Category", x = "Coaster Category")

coasterType <- ggplot(data=game_data, mapping = aes(x = fct_infreq(Coaster_Type)))+
```

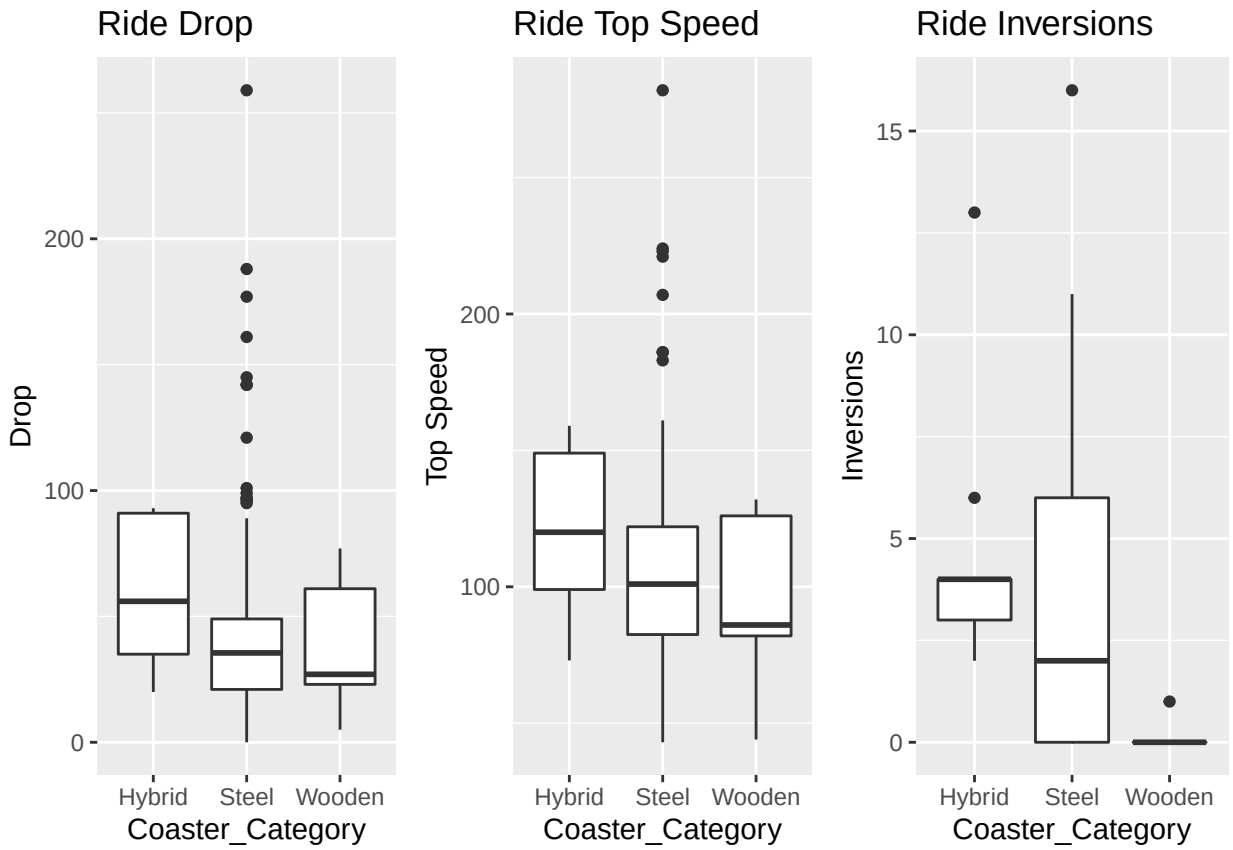
```
  geom_bar(fill="#FF6666")+
  coord_flip()+
  labs(title = "Coaster count per Type", x = "Coaster Type")

grid.arrange(coasterMainCat, coasterType, nrow=1)
```



Each coaster category has it's own features. We want to **compare drop, speed and number of inversions between the three main categories**. We see that **Hybrid coasters tend to be faster and we higher drops**, but some observation of Steel coasters are even faster and higher. **Wooden coasters have no inversions** except for one and **Steel coasters have the wider range in the number of inversions**.

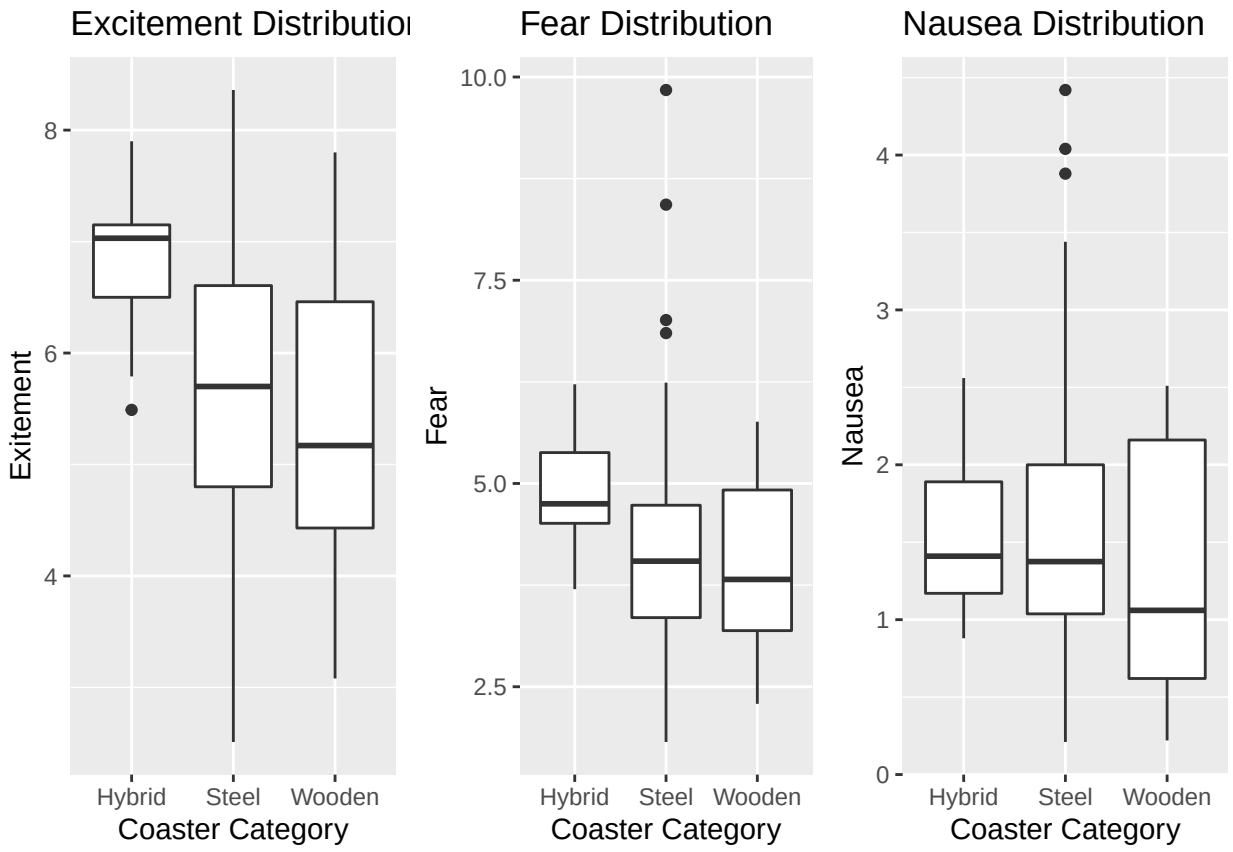
```
# box plots drop vs type
heightComp <- game_data %>%
  ggplot(aes(x=Coaster_Category, y=Biggest_Drop))+
  geom_boxplot()+
  labs(title = "Ride Drop", y="Drop")

# box plots speed vs type
speedComp <- game_data %>%
  ggplot(aes(x=Coaster_Category, y=Max_Speed))+
  geom_boxplot()+
  labs(title = "Ride Top Speed", y="Top Speed")

# box plots inversions vs type
invComp <- game_data %>%
  ggplot(aes(x=Coaster_Category, y=Inversions_Num))+
  geom_boxplot()+
```

```
  labs(title = "Ride Inversions", y="Inversions")

grid.arrange(heightComp, speedComp, invComp, nrow=1)
```



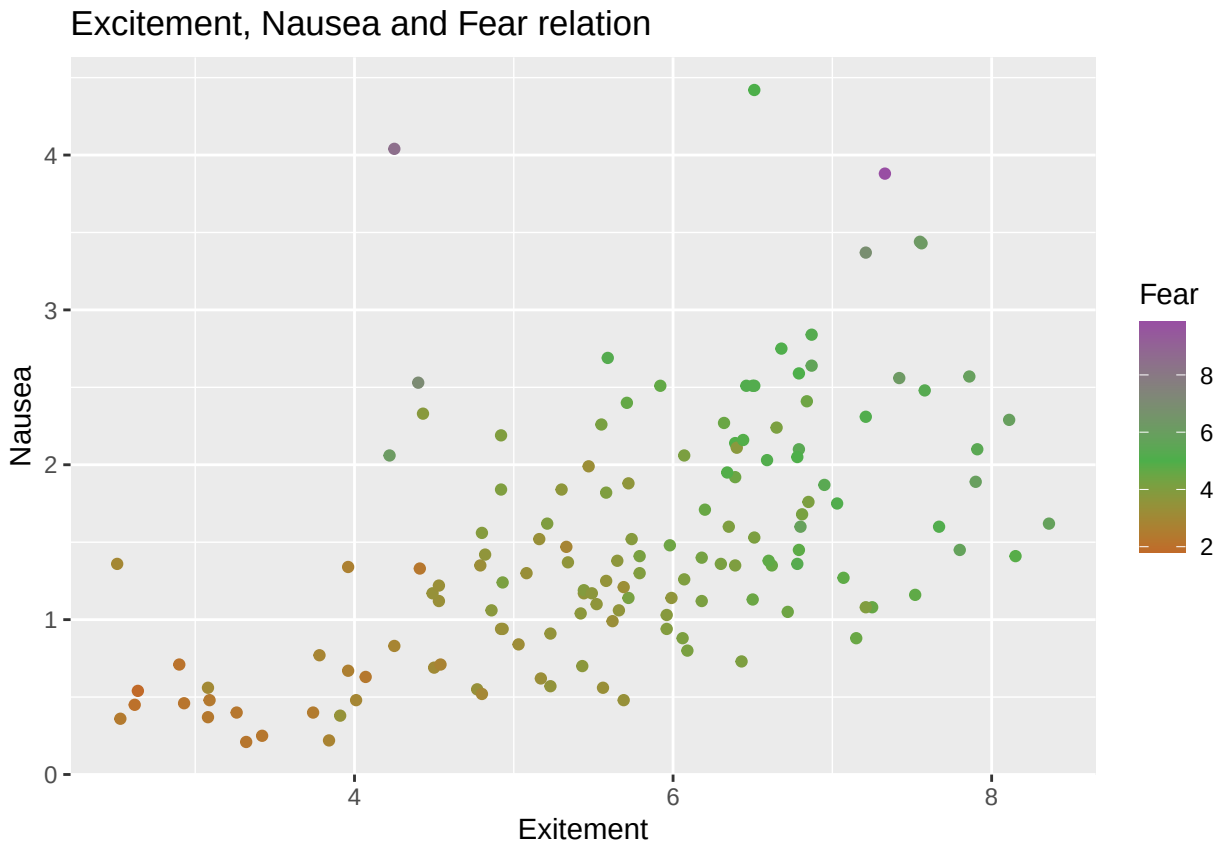
The following boxplots show the **distribution of the rides for each category** based on the three metrics of evaluation of a ride. Hybrid coasters seems to be the most exciting and also the most scaring. Steel coasters have the wider range of excitement, fear and nausea. **Basically hybrid coasters are very attractive and rides are similar to each other (homogeneous). Wooden and steel coasters can be both attracting and boring and are more varied.** There are a few observations of steel coasters which presents higher values of fear and nausea.

```
excitement <- ggplot(game_data, aes(x=Coaster_Category, y=Exitement))+
  geom_boxplot()+
  labs(title = "Excitement Distribution", x = "Coaster Category")

fear <- ggplot(game_data, aes(x=Coaster_Category, y=Fear))+
  geom_boxplot()+
  labs(title = "Fear Distribution", x = "Coaster Category")

nausea <- ggplot(game_data, aes(x=Coaster_Category, y=Nausea))+
  geom_boxplot()+
  labs(title = "Nausea Distribution", x = "Coaster Category")

grid.arrange(excitement, fear, nausea, nrow=1)
```



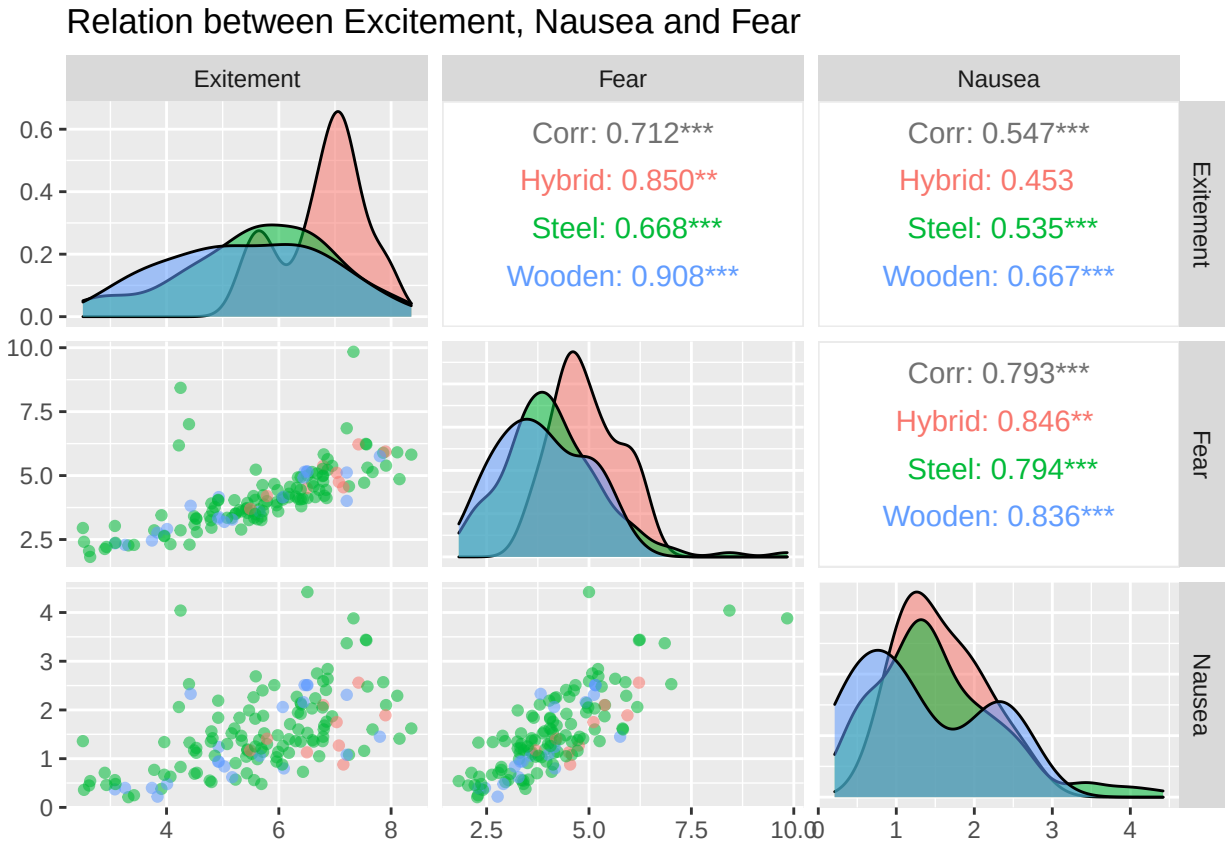
The three metrics, excitement, fear and nausea, **appear to be somehow related** as shows the following plot. In order to give a more precise analysis, let's compute also a pairplot.

```
ggplot(game_data, aes(x=Exitement, y=Nausea, color=Fear))+
  geom_point()+
  scale_color_gradient2(midpoint=5,low="#e41a1c",mid="#4daf4a",high="#984ea3",space="Lab")+
  labs(title = "Excitement, Nausea and Fear relation")
```



Looking at the following pairplot, it appears to be a **relation between excitement and fear**. Also **fear and nausea seems to be in some sort of relation**. The hypothesis here is that nausea affects the fear, and the fear affects the excitement. **Basically these three metrics are interrelated influencing each other.**

```
# pearson coefficient
# *** 99.9%, ** 99%, * 95%, . 90%, "" otherwise
ggpairs(game_data, columns = c("Exitement", "Fear", "Nausea"),
        aes(color=Coaster_Category, alpha=.5),
        title="Relation between Excitement, Nausea and Fear")
```

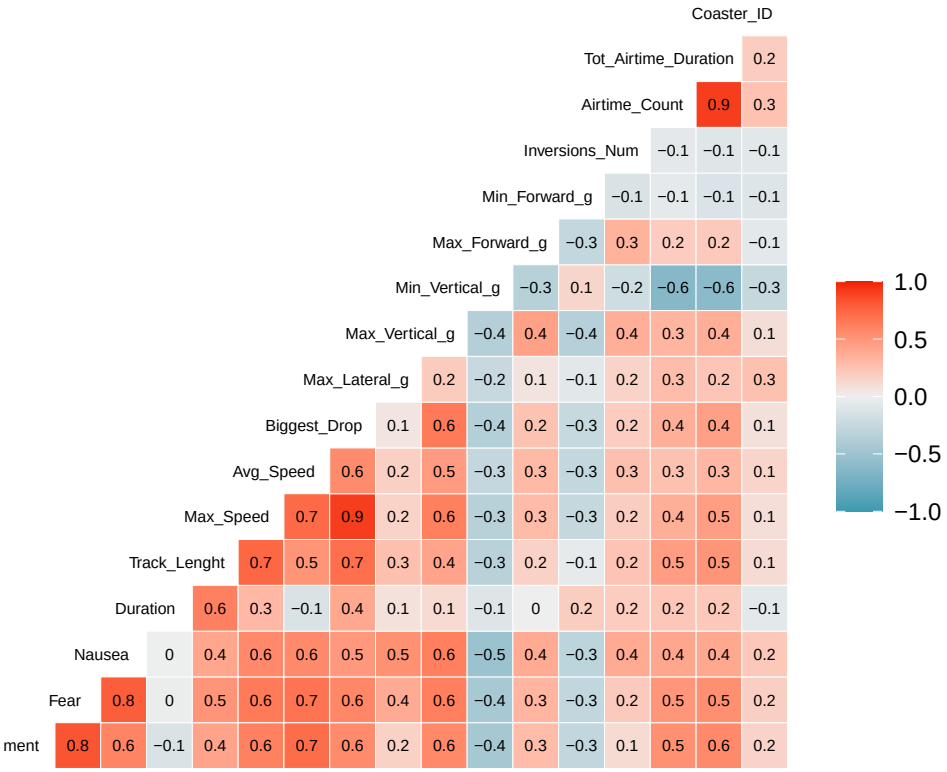


The above plot is done using the GGally package, an extension of ggplot2. The description and source code can be found at: link . There is also a pdf document that shows the complete usage of this package. It includes lots of interesting features and visualization options. The one reported here is just an example.

Now that we know how the metrics are related, it is interesting to see which of the technical specs of the rides influence the most this metrics. This means **identify the most important factors that determine a good or a bad ride**. In order to do so, we use a **correlation matrix** computed with the Spearman method, able to catch monotonic pairwise relations.

```
#correlation matrix numerical values
game_data%>%
  select(-Coaster_Type, -Launched, -Shuttle, -Coaster_Category, -coaster_class)%>%
  ggcorr(method=c("pairwise", "spearman"), label=TRUE, label_size = 2,
         legend.size = 9, size=2, hjust=.90)+
  labs(title = "Metrics Correlation Matrix")
```

Metrics Correlation Matrix



From this result we spot **some correlations**:

- excitement –> fear, nausea, max_speed, avg_speed, biggest_drop, max_vertical_g, tot_airtime_duration
- fear –> nausea, track_length, max_speed, avg_speed, biggest_drop, max_vertical_g
- nausea –> max_speed, avg_speed, max_lateral_g, max_vertical_g

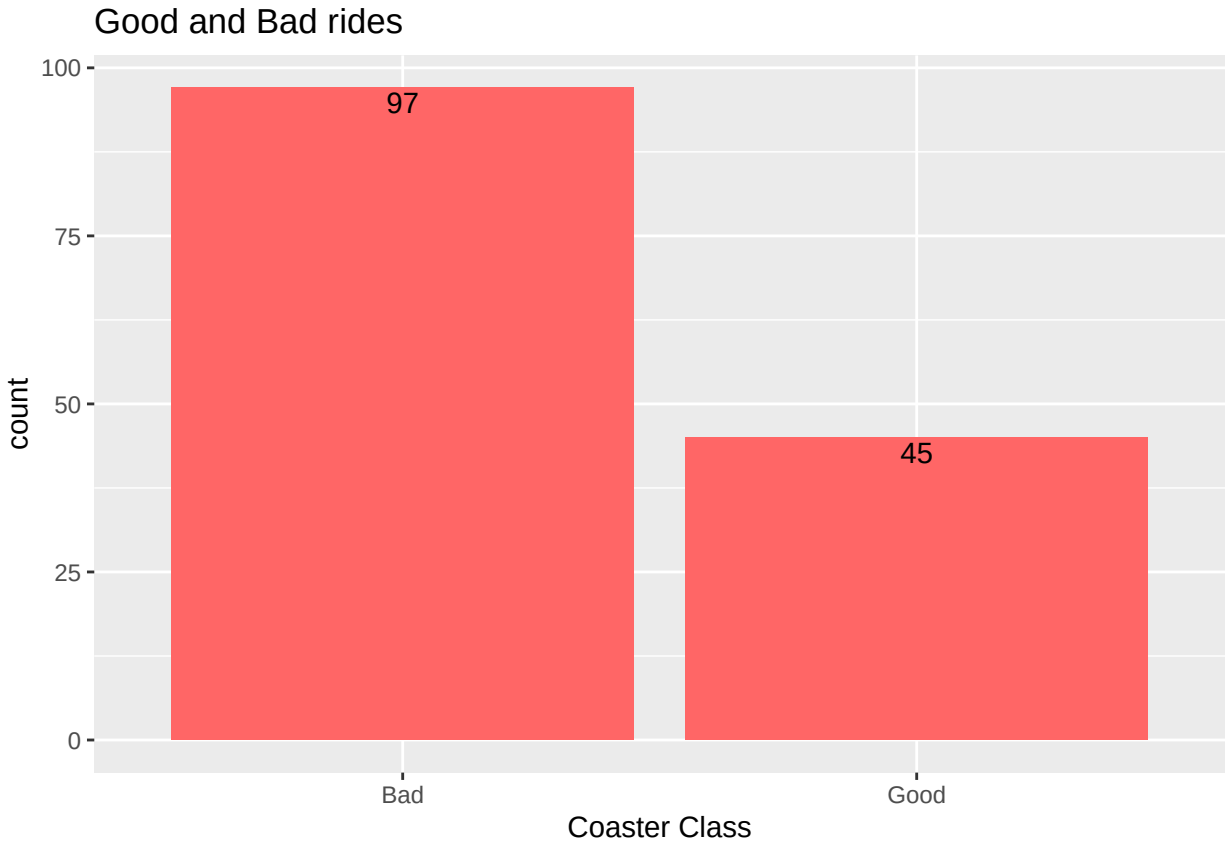
This features seems to be the **most related to the metrics**:

- Max_Speed, Avg_Speed, Biggest_Drop, Max_Vertical_g, Tot_Airtime_Duration

Basically tweaking this specs it is possible to obtain good rides.

Another important thing to visualize is the **number of good and bad rides present** in the data set, using the *coaster_class* attribute created in the preprocessing stage. We see that the bad class is the majority, meaning **how difficult and non-trivial is building good rides**.

```
#class balance
ggplot(game_data, aes(x=coaster_class))+
  geom_bar(fill="#FF6666")+
  geom_text(mapping = aes(label=..count..), stat = "count", vjust=1.2)+
  labs(title = "Good and Bad rides", x = "Coaster Class")
```



5.1 Game Ride Analysis: Conlusions

After this analysis we conclude that Steel coasters are most popular also in the game and hybrid coasters are the most extreme. **Basically hybrid coasters are very attractive and rides are similar to each other (homogeneous). Wooden and steel coasters can be both attracting and boring and are more varied.** We also looked at the coaster metrics and understood that **the three metrics are interrelated influencing each other.** At the end we analized the technical factor of a ride affecting the most the rating, and it emerged that speed, drop, vertical g forces and the air time duration are the most important attributes in determining the level of satisfaction of the riders of a ride. The difference in the number of good and bad rides allows us to realize **how difficult and non-trivial is building good rides**.

6. Game VS Reality: Roller Coasters Comparison

In this section i want to **compare game created and real rides to see the differences** and try to understand if game rides can exists in the real life. Let's start with a summary of the the technical details of the rides. First the real ones and then the game ones.

```
# summary of real rides
rides %>%
  filter(Ride_Type=="Real")%>%
  summary()
```

```
##    Max_Speed       Biggest_Drop     Track_Lenght       Duration
##  Min.   : 16.00   Min.   :  3.00   Min.   :  61.0   Min.   : 16.00
##  1st Qu.: 80.00   1st Qu.: 23.75   1st Qu.: 712.2   1st Qu.: 92.75
##  Median : 88.00   Median : 30.50   Median : 884.0   Median :120.00
##  Mean   : 92.35   Mean   : 35.91   Mean   : 964.6   Mean   :122.06
##  3rd Qu.:104.50   3rd Qu.: 44.00   3rd Qu.:1215.0   3rd Qu.:150.00
```

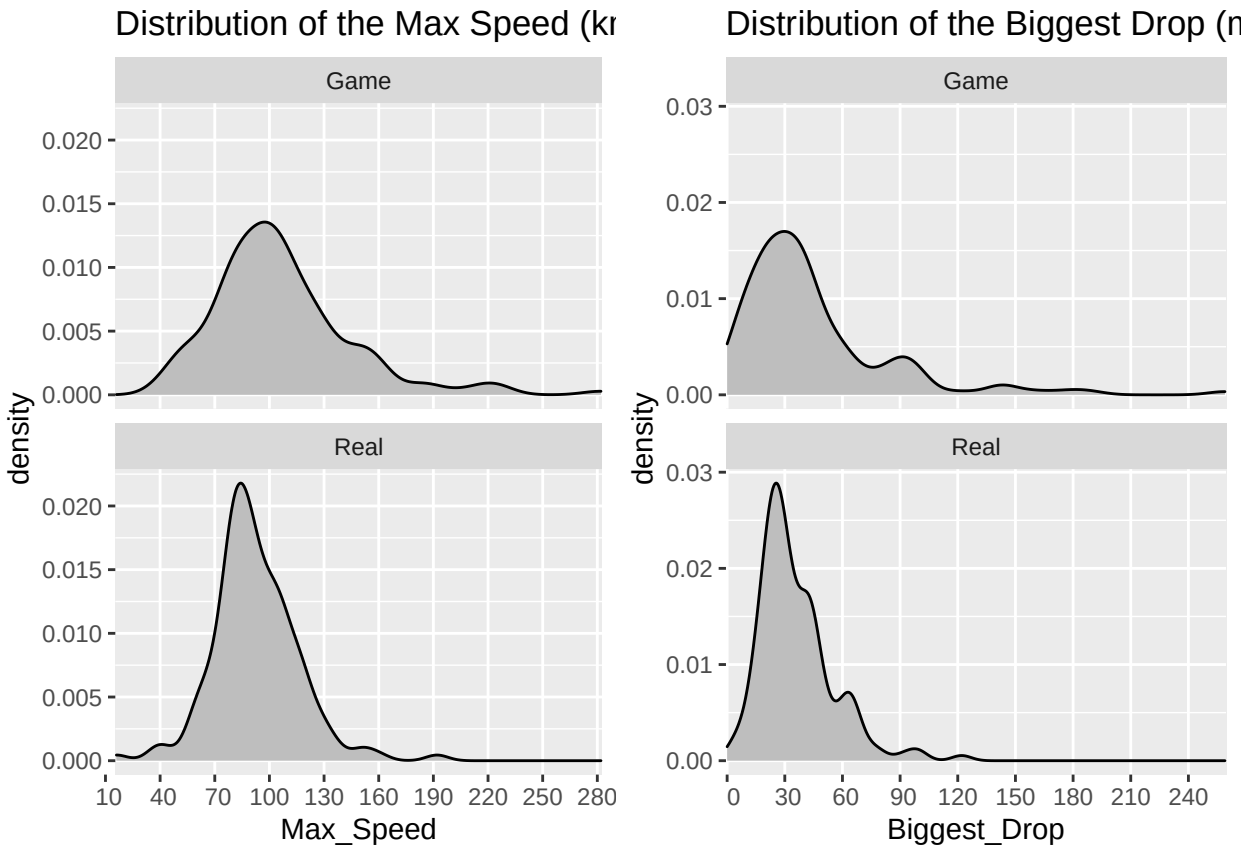
```
##  Max.   :192.00   Max.   :122.00   Max.   :2244.0   Max.   :250.00
##  Inversions_Num   Ride_Type
##  Min.   :0.000   Length:148
##  1st Qu.:0.000   Class :character
##  Median :0.000   Mode  :character
##  Mean   :1.601
##  3rd Qu.:3.000
##  Max.   :7.000
```

```
# summary of game rides
rides %>%
  filter(Ride_Type=="Game")%>%
  summary()
```

```
##    Max_Speed       Biggest_Drop     Track_Lenght       Duration
##  Min.   : 43.00   Min.   :  0.00   Min.   : 176.0   Min.   : 23.00
##  1st Qu.: 82.25   1st Qu.: 21.25   1st Qu.: 604.2   1st Qu.: 71.25
##  Median :101.00   Median : 35.00   Median : 830.5   Median : 94.75
##  Mean   :106.87   Mean   : 44.92   Mean   : 992.0   Mean   :101.45
##  3rd Qu.:122.75   3rd Qu.: 55.50   3rd Qu.:1288.0   3rd Qu.:121.88
##  Max.   :282.00   Max.   :259.00   Max.   :3029.0   Max.   :299.20
##  Inversions_Num    Ride_Type
##  Min.   : 0.000   Length:142
##  1st Qu.: 0.000   Class :character
##  Median : 2.000   Mode  :character
##  Mean   : 2.866
##  3rd Qu.: 5.000
##  Max.   :16.000
```

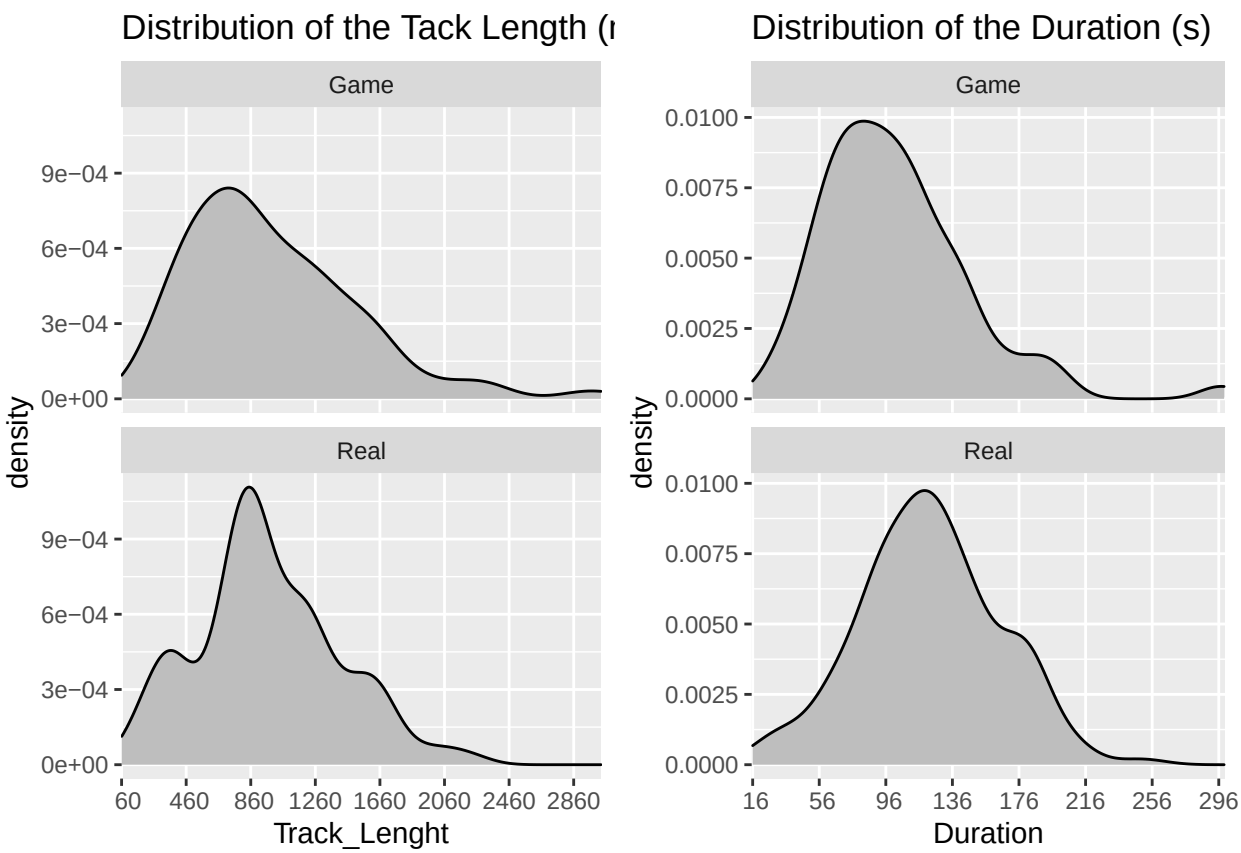
It seems that **game rides tend to be faster, higher, longer and with more inversion** the the real ones. Let's check this assumption with some visualizations. The following plots show the distribution of the max speed and the biggest drop for real and game rides. **Real rides values are concentrated in a narrow range and tend to have lower values**. Game rides are more varied and tend to have higher values.

```
# max speed distribution
speed_density <- ggplot(rides, aes(x=Max_Speed))+
  geom_density(fill="gray")+
  facet_wrap(vars(factor(Ride_Type)), ncol = 1)+
  scale_x_discrete(limits=seq(10, 282, 30))+
  labs(title = "Distribution of the Max Speed (km/h)")

# drop distribution
drop_density <- ggplot(rides, aes(x=Biggest_Drop))+
  geom_density(fill="gray")+
  facet_wrap(vars(factor(Ride_Type)), ncol = 1)+
  scale_x_discrete(limits=seq(0, 259, 30))+
  labs(title = "Distribution of the Biggest Drop (m)")

grid.arrange(speed_density, drop_density, nrow=1)
```



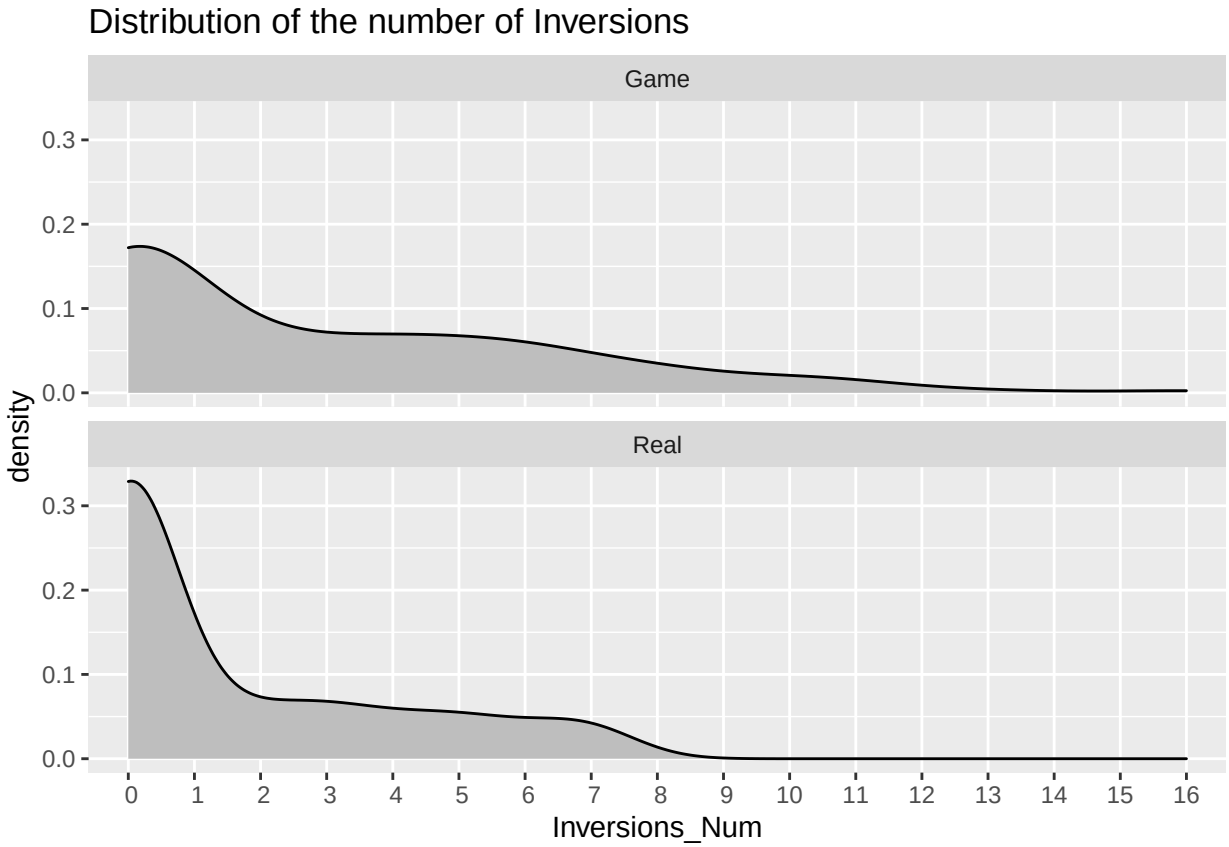
Regarding the track length, game and real rides are similar, but also in this case **game rides have higher values**. **Duration, however, is higher for the real rides**.

```
# length distribution
length_density <- ggplot(rides, aes(x=Track_Lenght))+
  geom_density(fill="gray")+
  facet_wrap(vars(factor(Ride_Type)), ncol = 1)+
  scale_x_discrete(limits=seq(60, 3029, 400))+
  labs(title = "Distribution of the Tack Length (m)")

# duration distribution
duration_density <- ggplot(rides, aes(x=Duration))+
  geom_density(fill="gray")+
  facet_wrap(vars(factor(Ride_Type)), ncol = 1)+
  scale_x_discrete(limits=seq(16, 300, 40))+
  labs(title = "Distribution of the Duration (s)")

grid.arrange(length_density, duration_density, nrow=1)
```



Also the number of inversion is **higher for the game rides.** In the real rides there is a clear cut on the number of inversions, remember from the summary that 75% of the rides have three or less than three inversions.

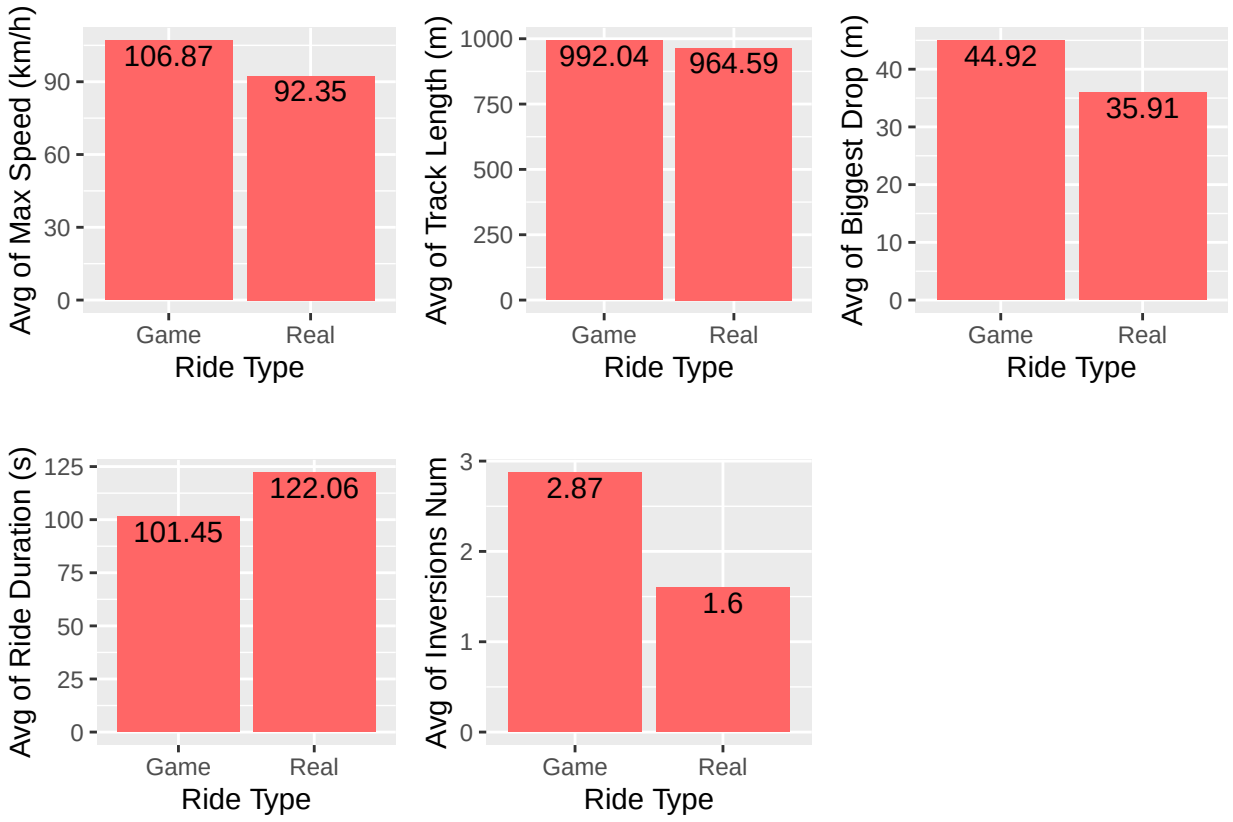
```
# inversions distribution
inversions_density <- ggplot(rides, aes(x=Inversions_Num))+
  geom_density(fill="gray")+
  facet_wrap(vars(factor(Ride_Type)), ncol = 1)+
  scale_x_discrete(limits=seq(0, 16, 1))+
  labs(title = "Distribution of the number of Inversions")

inversions_density
```



After the distributions, we look at the **average values** of the same attributes to confirm what observed.

```
# avg speed for coaster cat
avgSpeed <- rides %>%
  group_by(Ride_Type)%>%
  summarise(avgSpeed=round(mean(Max_Speed), digits = 2))%>%
  ggplot(aes(x=Ride_Type, y=avgSpeed))+
    geom_col(fill="#FF6666")+
    geom_text(mapping = aes(label=avgSpeed), vjust=1.2)+
    labs(title = "", x = "Ride Type", y="Avg of Max Speed (km/h)")

# avg drop for coaster cat
avgDrop <- rides %>%
  group_by(Ride_Type)%>%
  summarise(avgDrop=round(mean(Biggest_Drop), digits = 2))%>%
  ggplot(aes(x=Ride_Type, y=avgDrop))+
    geom_col(fill="#FF6666")+
    geom_text(mapping = aes(label=avgDrop), vjust=1.2)+
    labs(title = "", x = "Ride Type", y="Avg of Biggest Drop (m)")

# avg duration for coaster cat
avgDuration <- rides %>%
  group_by(Ride_Type)%>%
  summarise(avgDuration=round(mean(Duration), digits = 2))%>%
  ggplot(aes(x=Ride_Type, y=avgDuration))+
    geom_col(fill="#FF6666")+
    geom_text(mapping = aes(label=avgDuration), vjust=1.2)+
```

```
    labs(title = "", x = "Ride Type", y="Avg of Ride Duration (s)")

# avg inversion for coaster cat
avgInv <- rides %>%
  group_by(Ride_Type)%>%
  summarise(avgInv=round(mean(Inversions_Num), digits = 2))%>%
  ggplot(aes(x=Ride_Type, y=avgInv))+
    geom_col(fill="#FF6666")+
    geom_text(mapping = aes(label=avgInv), vjust=1.2)+
    labs(title = "", x = "Ride Type", y="Avg of Inversions Num")

# avg length for coaster cat
avgLnt <- rides %>%
  group_by(Ride_Type)%>%
  summarise(avgLnt=round(mean(Track_Lenght), digits = 2))%>%
  ggplot(aes(x=Ride_Type, y=avgLnt))+
    geom_col(fill="#FF6666")+
    geom_text(mapping = aes(label=avgLnt), vjust=1.2)+
    labs(title = "", x = "Ride Type", y="Avg of Track Length (m)")

grid.arrange(avgSpeed, avgLnt, avgDrop, avgDuration, avgInv, nrow=2)
```

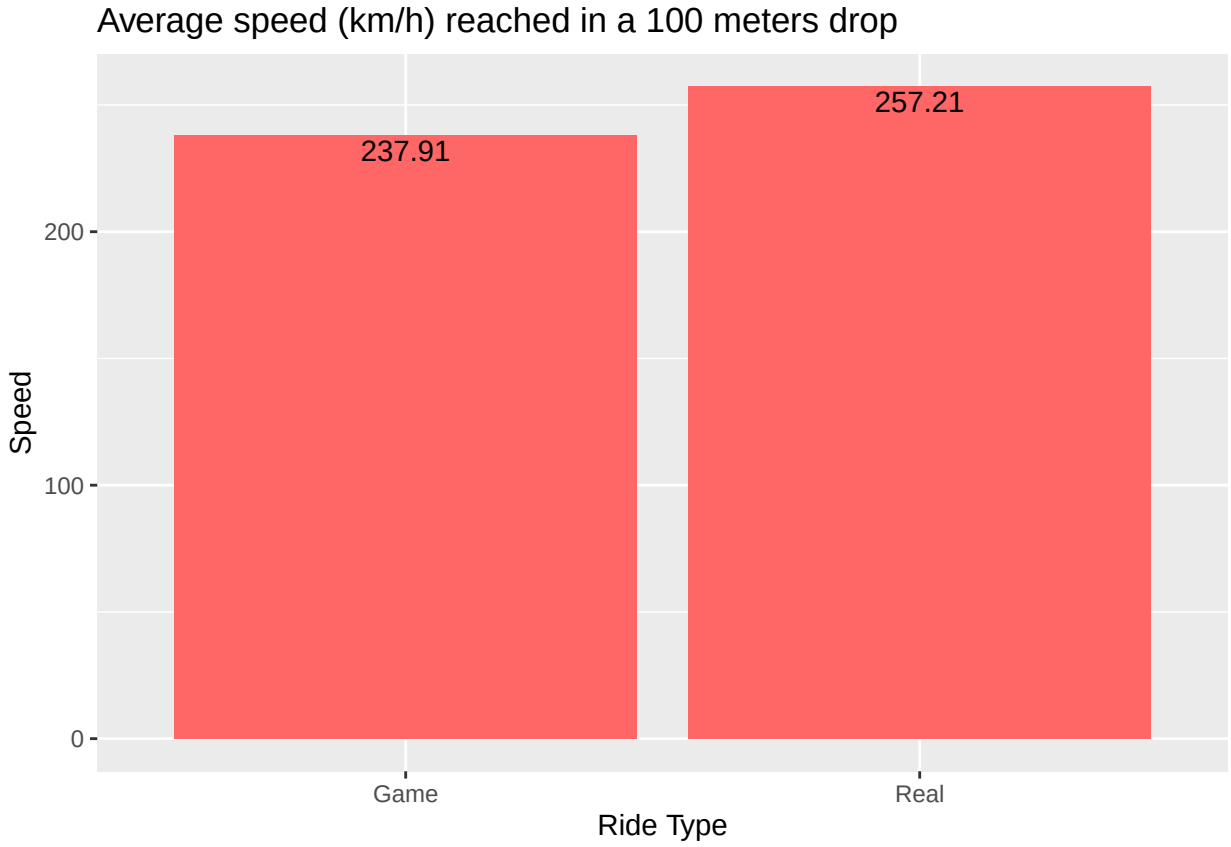


As we see in the plots above, **average values for real and game rides are not that different**. Real life rides are a little bit shorter and slower, but last longer in terms of duration in seconds.

Now i want to see if the physics in the game is realistic by analyzing **how the drop influences the speed**. First we compute the average drop and average speed for real and game rides, then we compute the ratio between this values. In real life coaster a shorter drop is needed to achieve certain speed –> it is as if the

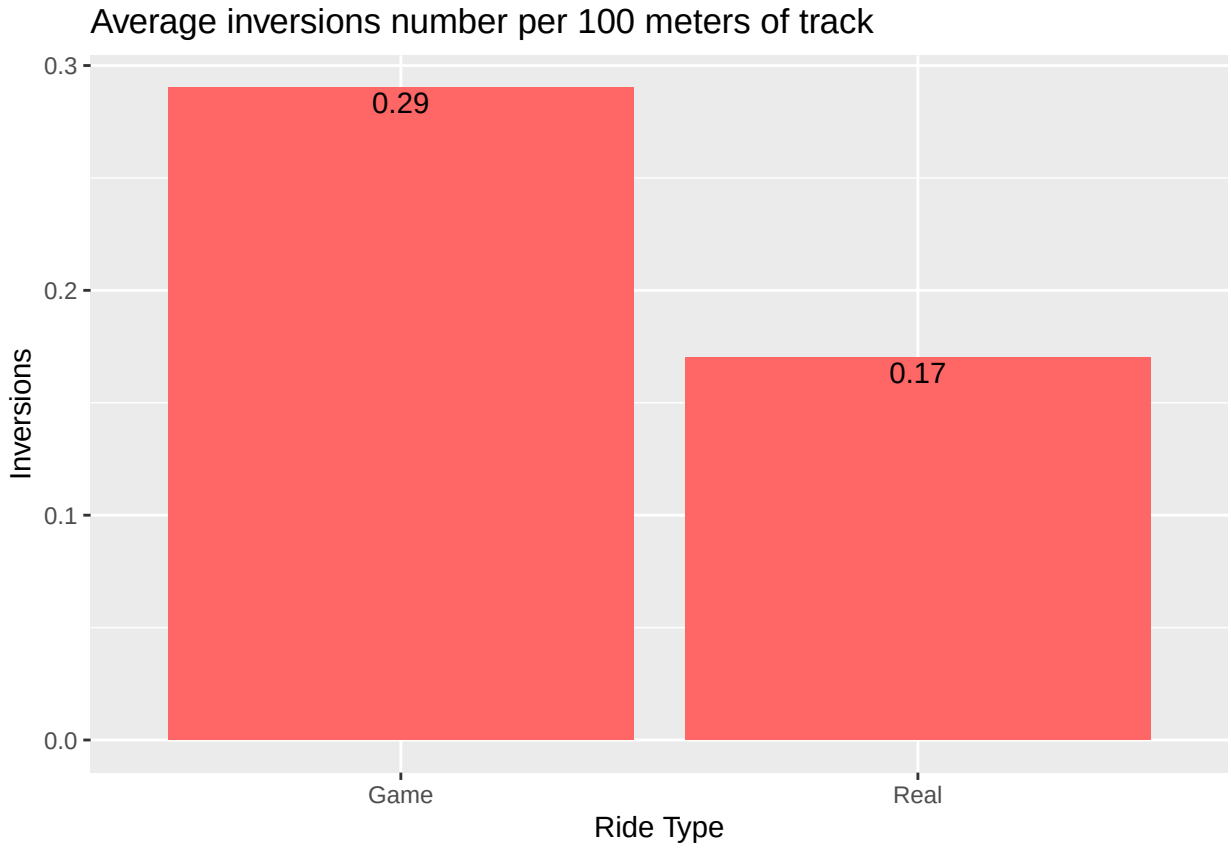
game coasters are somehow "slowed down". **Basically game rides accelerate a little bit slower on drops.**

```
# ratio between drop and speed
rides%>%
  group_by(Ride_Type)%>%
  summarise(meanDrop = mean(Biggest_Drop), meanSpeed = mean(Max_Speed))%>%
  mutate(dropSpeedRatio = round(meanSpeed/(meanDrop/100), digits = 2))%>% # for each 100m
  ggplot(aes(y=dropSpeedRatio, x=Ride_Type))+
    geom_col(fill="#FF6666")+
    geom_text(mapping = aes(label=dropSpeedRatio), vjust=1.2)+
    labs(title = "Average speed (km/h) reached in a 100 meters drop",
         x = "Ride Type", y="Speed")
```



With the following plot, i want to compare the **"density" of inversions** in the rides by dividing the number of inversions by the length of the track. Game rides have a higher concentration of inversions.

```
# inversion in track length
rides%>%
  group_by(Ride_Type)%>%
  summarise(meanInv = mean(Inversions_Num), meanLnt = mean(Track_Lenght))%>%
  mutate(invLengthRatio = round(meanInv/(meanLnt/100), digits = 2))%>% # for each 100m
  ggplot(aes(y=invLengthRatio, x=Ride_Type))+
    geom_col(fill="#FF6666")+
    geom_text(mapping = aes(label=invLengthRatio), vjust=1.2)+
    labs(title = "Average inversions number per 100 meters of track",
         x = "Ride Type", y="Inversions")
```



6.1 Game vs Reality: Rides Comparison Conslusions

We can say that **game rides tend to be faster, higher, longer and with more inversion** in respect to real rides, and also are **more varied** than real rides, that means are very different from each other. We observed a **similar shape in the distribution of the technical specs**, especially the duration. **Real life however last longer** in terms of duration in seconds. We also understood the **game physics works well**, ride behavior is similar to reality although **game rides accelerate a little bit slower** on the drops (this can be overcome by tweaking a friction parameter in the game). We can conclude our analysis by stating that **Game and Real rides are not so different**, this means that game rides could potentially exist in real life.

7. Drawing Conclusions - Summary

In this project i tried to **analyze different aspect related to roller coasters** and theme parks. Starting with the **cleaning and preprocessing** pahses we fix some issues with the data sets: **null values are dropped** and the **types of the columns** is changed properly. Also we removed some columns not important for the analysis and **converted the units** from imperial to metrics. The preprocessing step ends with the **creation of a new data set**, obtained merging together the common columns of the other two data sets. After this preliminary step, we go on with an **analysis of real existing rides** we gained some knowledge about the **evolution of the roller coaster** rides and the **roller coaster industry over the years**. We also figured out **how rides and theme parks are distributed over the US** and why some states have the majority of parks, with the use of maps. After that we change the context and **analyzed rides created in a popular game: Planet Coaster**, noticing that some of the observation we did for real rides are still true. We also have now a clear idea of the **factor that determines a ride that guests will like.** In the last part we **exposed the differences between real and game rides**, noticing that reality in this case is not that far from the game and the **rides built in the game could potentially be built in real life**. I'm satisfied with the obtained results and with what i observed.

Edoardo Bianchi, 2022