Project

```r
# Load libraries
library(ggplot2)
library(dplyr)
library(lubridate)
library(tidyr)

# Read the data from CSV
data <- read.csv("data.csv")

# Convert Date to Date format
data$Date <- as.Date(data$Date)

# Extract month and year for time-based analysis
data$Year <- year(data$Date)
data$Month <- month(data$Date)
data$Season <- case_when(
  data$Month %in% 3:5  ~ "Spring",
  data$Month %in% 6:8  ~ "Summer",
  data$Month %in% 9:11 ~ "Autumn",
  TRUE                 ~ "Winter"
)
# For temperature, you may need to clean/remove the "C" symbol and convert them to numeric values
data$Day.Temp <- as.numeric(gsub("C", "", data$Day.Temp))
data$Night.Temp <- as.numeric(gsub("C", "", data$Night.Temp))


data <- data %>%
  filter(Day.Weather != "" & Day.Weather != "-") %>%
  filter(Night.Weather != "" & Night.Weather != "-") %>%
  filter(Day.Wind.Force != "Unknown" & Night.Wind.Force != "Unknown")


# Function to categorize wind force levels
categorize_wind_force <- function(force) {
  if (force == "Unknown") {
    return("0")
  } else if (grepl(" 3 |3 |1-2 ", force)) {
    return("0-3")
  } else if (grepl("3-4 |4 |4-5 |5-6 |5 |6 ", force)) {
    return("3-6")
  } else if (grepl("6-7 |8-9 |7-8 ", force)) {
    return("6-9")
  } else if (grepl("9-10 |10-11 |11-12 ", force)) {
    return("9-12")
  } else {
    return(NA)
```

```
  }
}

# Apply the function to both Day_Wind_Force and Night_Wind_Force columns
data <- data %>%
  mutate(
    Day_Wind_Force_Category = sapply(Day.Wind.Force, categorize_wind_force),
    Night_Wind_Force_Category = sapply(Night.Wind.Force, categorize_wind_force)
  )

# View the updated dataframe
print(head(data))
```

```
##         Date    City Weather.Condition Temperature
## 1 2011-01-01 beijing              /       0℃/-9℃
## 2 2011-01-02 beijing             /      -2℃/-7℃
## 3 2011-01-03 beijing              /       1℃/-8℃
## 4 2011-01-04 beijing              /     -1℃/-11℃
## 5 2011-01-05 beijing              /      -1℃/-8℃
## 6 2011-01-06 beijing              /      0℃/-10℃
##        Wind.Force.and.Direction Day.Weather Night.Weather Day.Temp Night.Temp
## 1    3 /   3                                  0         -9
## 2    3 /   3                                -2         -7
## 3       3-4 /   3                                   1         -8
## 4    3 /   3                                 -1        -11
## 5           4-5 / 3-4                                  -1         -8
## 6    3 /   3                                  0        -10
##   Day.Wind.Direction Night.Wind.Direction Day.Wind.Force Night.Wind.Force Year
## 1                                       3              3  2011
## 2                                       3              3  2011
## 3                                     3-4                3  2011
## 4                                       3              3  2011
## 5                                             4-5              3-4  2011
## 6                                       3              3  2011
##   Month Season Day_Wind_Force_Category Night_Wind_Force_Category
## 1     1 Winter                     0-3                       0-3
## 2     1 Winter                     0-3                       0-3
## 3     1 Winter                     3-6                       0-3
## 4     1 Winter                     0-3                       0-3
## 5     1 Winter                     3-6                       3-6
## 6     1 Winter                     0-3                       0-3
```

```
# Summary statistics for temperature (Day and Night)
summary_stats <- data %>%
  group_by(City) %>%
  summarise(
    Day_Temperature_Mean = mean(Day.Temp, na.rm = TRUE),
    Night_Temperature_Mean = mean(Night.Temp, na.rm = TRUE),
    Day_Temperature_Median = median(Day.Temp, na.rm = TRUE),
    Night_Temperature_Median = median(Night.Temp, na.rm = TRUE),
    Day_Temperature_Min = min(Day.Temp, na.rm = TRUE),
    Night_Temperature_Min = min(Night.Temp, na.rm = TRUE),
    Day_Temperature_Max = max(Day.Temp, na.rm = TRUE),
```

```
    Night_Temperature_Max = max(Night.Temp, na.rm = TRUE)
  )
print(summary_stats)
```

```
## # A tibble: 4 x 9
##   City     Day_Temperature_Mean Night_Temperature_Mean Day_Temperature_Median
##   <chr>                   <dbl>                  <dbl>                    <dbl>
## 1 beijing                  18.9                   8.55                       20
## 2 haikou                   28.6                  22.2                        30
## 3 lasa                     17.1                   3.43                       17
## 4 shanghai                 21.2                  14.9                        22
## # i 5 more variables: Night_Temperature_Median <dbl>,
## #   Day_Temperature_Min <dbl>, Night_Temperature_Min <dbl>,
## #   Day_Temperature_Max <dbl>, Night_Temperature_Max <dbl>
```

```
# Combine both Day and Night wind force categories into one column for analysis
df_long <- data %>%
  gather(key = "Time_of_Day", value = "Wind_Force_Category", Day_Wind_Force_Category, Night_Wind_Force_C

# Summary statistics: count of wind force categories per city
summary_stats <- df_long %>%
  group_by(City, Wind_Force_Category) %>%
  summarise(Count = n()) %>%
  arrange(City, Wind_Force_Category)

# Print summary statistics
print(summary_stats)
```

```
## # A tibble: 13 x 3
## # Groups:   City [4]
##    City     Wind_Force_Category Count
##    <chr>    <chr>               <int>
##  1 beijing  0-3                  5634
##  2 beijing  3-6                  1622
##  3 beijing  6-9                     2
##  4 haikou   0-3                  2571
##  5 haikou   3-6                  4671
##  6 haikou   6-9                    36
##  7 haikou   9-12                   14
##  8 lasa     0-3                  6136
##  9 lasa     3-6                   621
## 10 lasa     6-9                     1
## 11 shanghai 0-3                  2994
## 12 shanghai 3-6                  4273
## 13 shanghai 6-9                    21
```

```
# Wind force summary statistics
wind_stats <- data %>%
  group_by(City) %>%
  summarise(
    Day_Wind_Force_Mean = mean(Day.Wind.Force, na.rm = TRUE),
    Night_Wind_Force_Mean = mean(Night.Wind.Force, na.rm = TRUE),
```

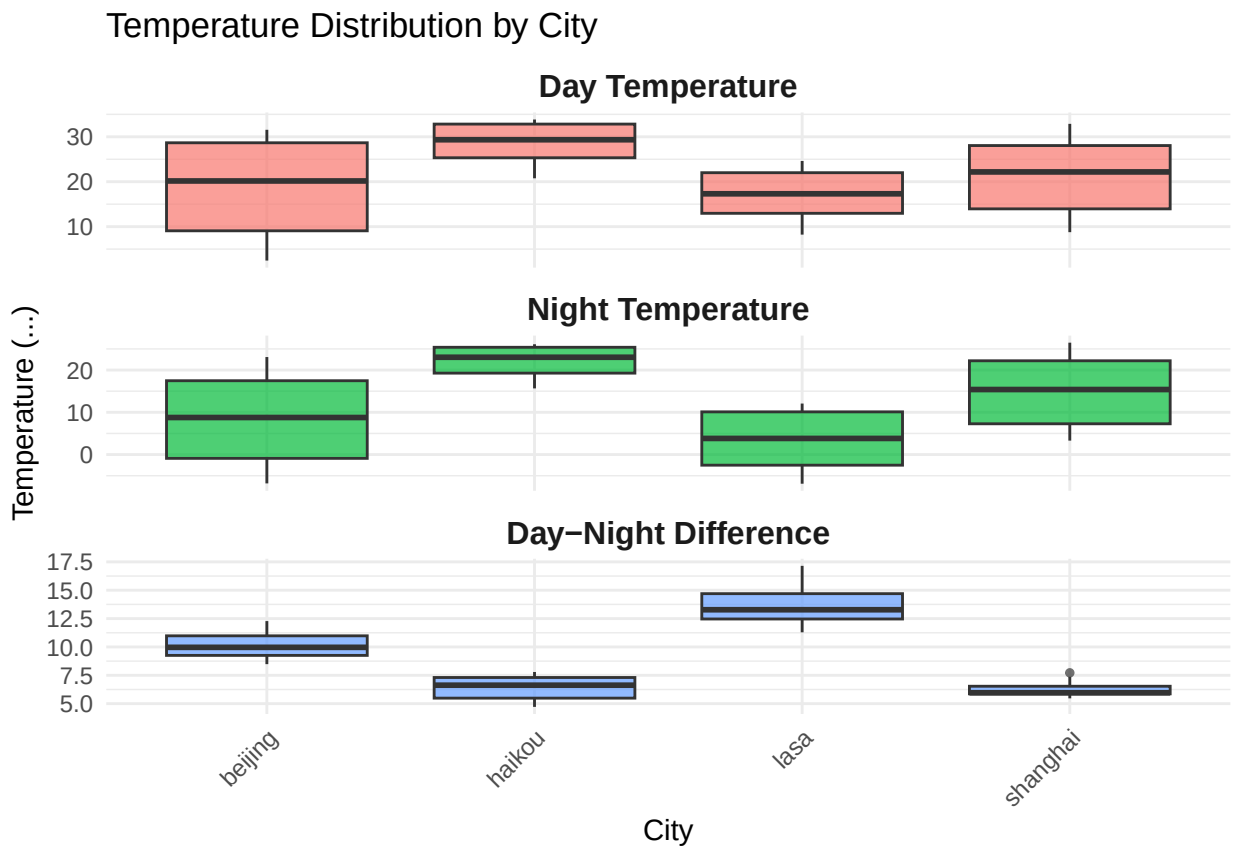
```
    Day_Wind_Force_Max = max(Day.Wind.Force, na.rm = TRUE),
    Night_Wind_Force_Max = max(Night.Wind.Force, na.rm = TRUE)
  )
print(wind_stats)
```

```
## # A tibble: 4 x 5
##   City     Day_Wind_Force_Mean Night_Wind_Force_Mean Day_Wind_Force_Max
##   <chr>                  <dbl>                 <dbl> <chr>
## 1 beijing                   NA                    NA 6
## 2 haikou                    NA                    NA 9-10
## 3 lasa                      NA                    NA 6
## 4 shanghai                  NA                    NA 8-9
## # i 1 more variable: Night_Wind_Force_Max <chr>
```

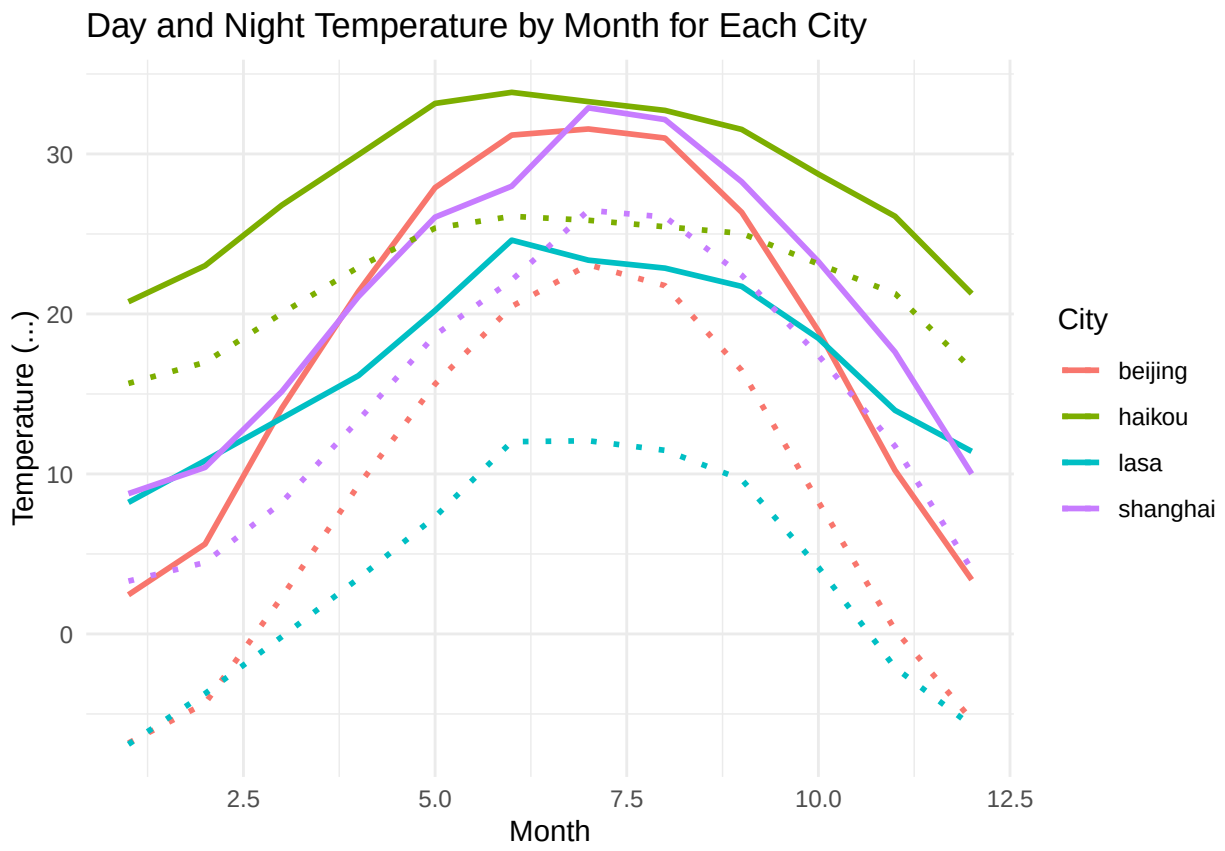
```
temperature_by_month <- data %>%
  group_by(City, Month) %>%
  summarise(
    Average_Day_Temperature = mean(Day.Temp, na.rm = TRUE),
    Average_Night_Temperature = mean(Night.Temp, na.rm = TRUE)
  ) %>%
  mutate(Day_Night_Temp_Difference = Average_Day_Temperature - Average_Night_Temperature)

# Calculate yearly averages and temperature difference
temperature_by_year <- data %>%
  group_by(City, Year) %>%
  summarise(
    Average_Day_Temperature = mean(Day.Temp, na.rm = TRUE),
    Average_Night_Temperature = mean(Night.Temp, na.rm = TRUE)
  ) %>%
  mutate(Day_Night_Temp_Difference = Average_Day_Temperature - Average_Night_Temperature)
```

```
# Reshape temperature_by_month for plotting
temperature_long_city <- temperature_by_month %>%
  pivot_longer(
    cols = c(Average_Day_Temperature, Average_Night_Temperature, Day_Night_Temp_Difference),
    names_to = "Temperature_Type",
    values_to = "Temperature"
  )
# Boxplots for Day, Night, and Difference by City
ggplot(temperature_long_city, aes(x = City, y = Temperature, fill = Temperature_Type)) +
  geom_boxplot(alpha = 0.7, outlier.size = 1) +
  facet_wrap(~ Temperature_Type, scales = "free_y", nrow = 3,
             labeller = labeller(Temperature_Type = c(
               "Average_Day_Temperature" = "Day Temperature",
               "Average_Night_Temperature" = "Night Temperature",
               "Day_Night_Temp_Difference" = "Day-Night Difference"
             ))) +
  labs(
    title = "Temperature Distribution by City",
    x = "City",
    y = "Temperature (℃)"
```

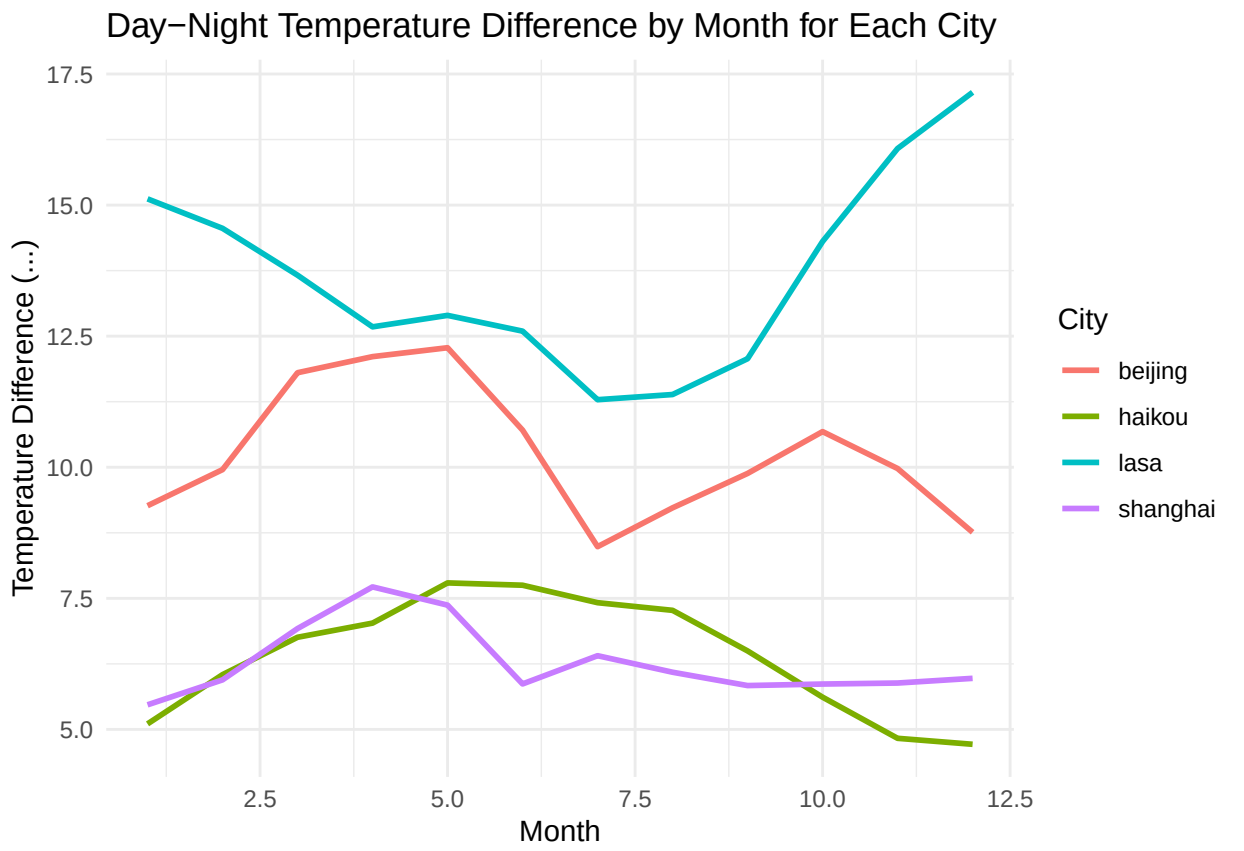
```
  ) +
  theme_minimal() +
  theme(
    legend.position = "none",
    strip.text = element_text(size = 12, face = "bold"),
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```



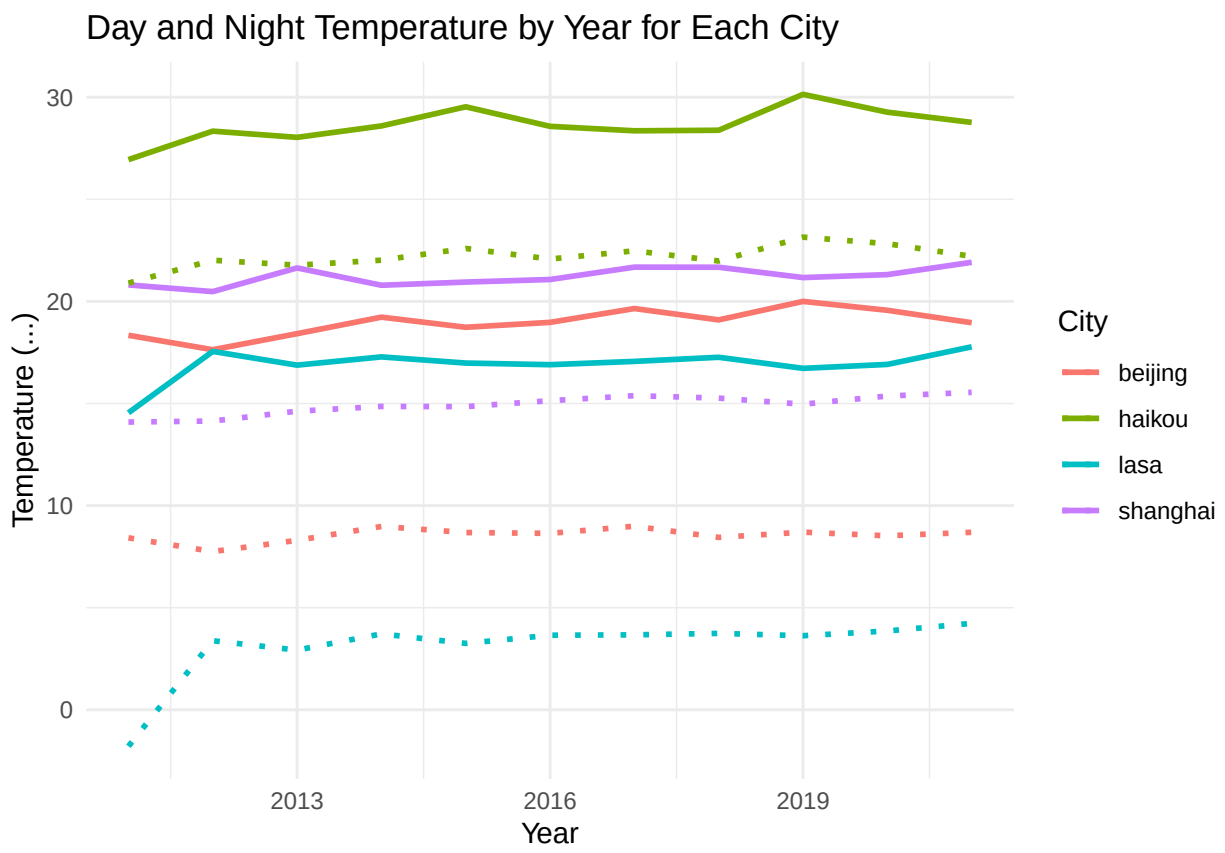

```
# Plot Day and Night Temperature by Month
ggplot(temperature_by_month) +
  geom_line(aes(x = Month, y = Average_Day_Temperature, color = City), linetype = "solid", size = 1) +
  geom_line(aes(x = Month, y = Average_Night_Temperature, color = City), linetype = "dotted", size = 1)
  labs(
    title = "Day and Night Temperature by Month for Each City",
    x = "Month",
    y = "Temperature (℃)",
    color = "City"
  ) +
  theme_minimal()
```



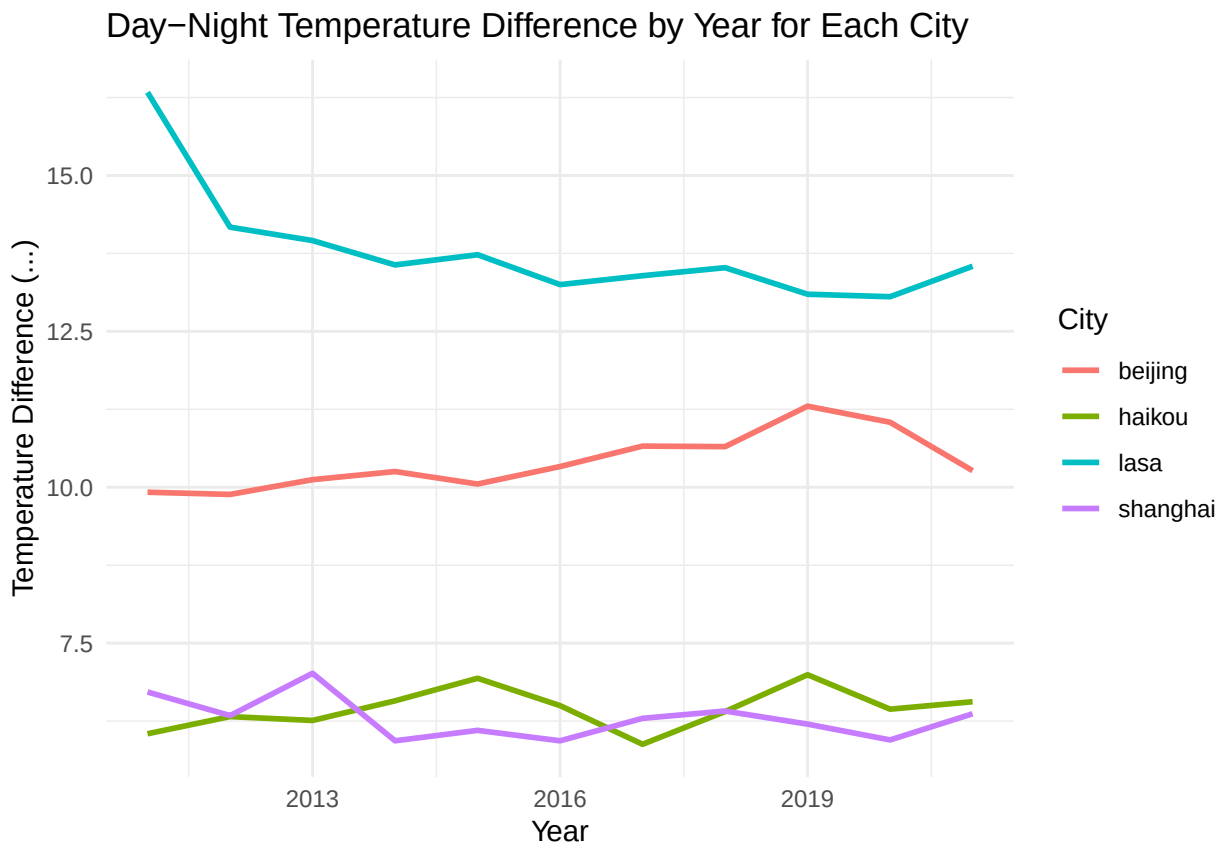

```r
# Plot Day-Night Temperature Difference by Month
ggplot(temperature_by_month, aes(x = Month, y = Day_Night_Temp_Difference, color = City)) +
  geom_line(size = 1) +
  labs(
    title = "Day-Night Temperature Difference by Month for Each City",
    x = "Month",
    y = "Temperature Difference (℃)",
    color = "City"
  ) +
  theme_minimal()
```




```r
# Plot Day and Night Temperature by Year
ggplot(temperature_by_year) +
  geom_line(aes(x = Year, y = Average_Day_Temperature, color = City), linetype = "solid", size = 1) +
  geom_line(aes(x = Year, y = Average_Night_Temperature, color = City), linetype = "dotted", size = 1) +
  labs(
    title = "Day and Night Temperature by Year for Each City",
    x = "Year",
    y = "Temperature (℃)",
    color = "City"
  ) +
  theme_minimal()
```

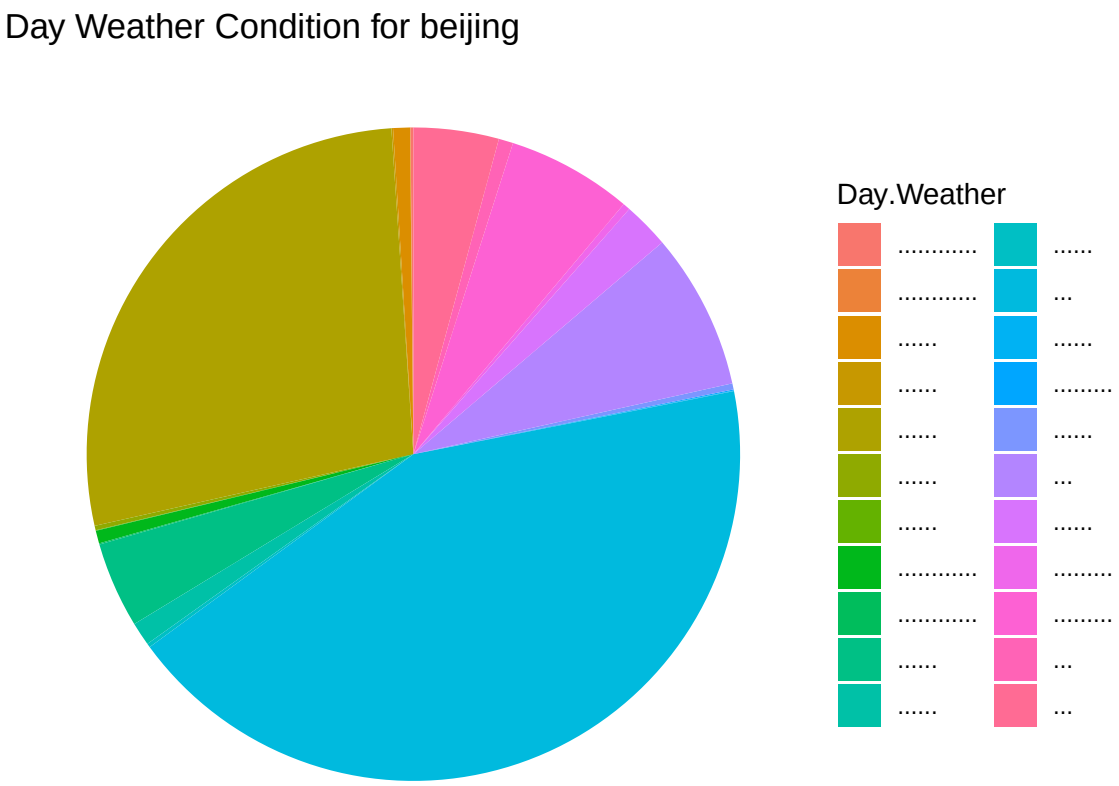



```r
# Plot Day-Night Temperature Difference by Year
ggplot(temperature_by_year, aes(x = Year, y = Day_Night_Temp_Difference, color = City)) +
  geom_line(size = 1) +
  labs(
    title = "Day-Night Temperature Difference by Year for Each City",
    x = "Year",
    y = "Temperature Difference (℃)",
    color = "City"
  ) +
  theme_minimal()
```

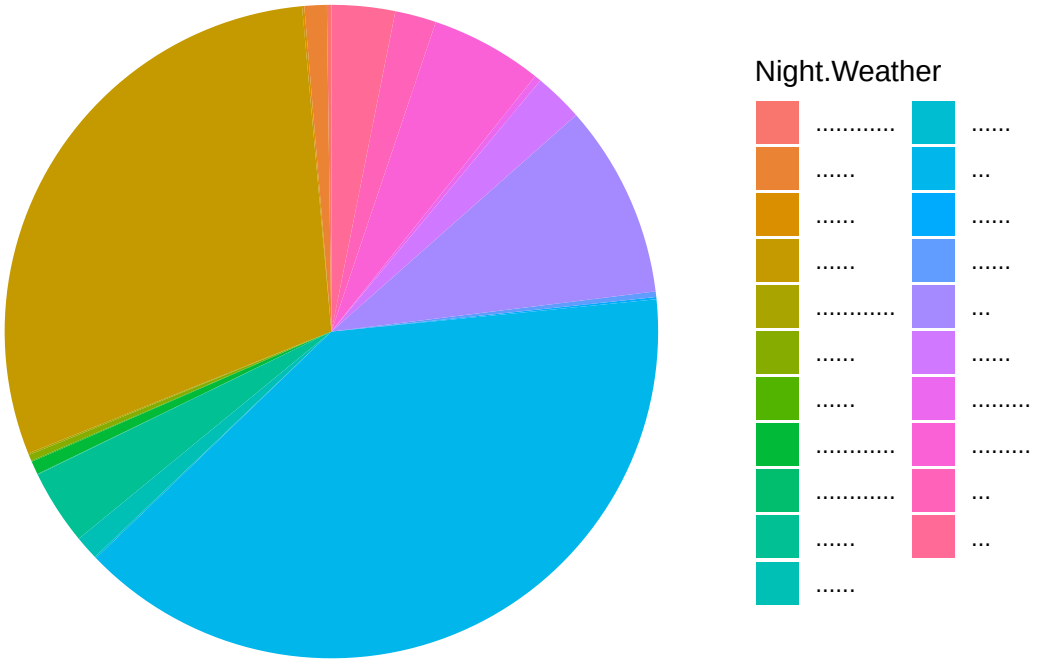


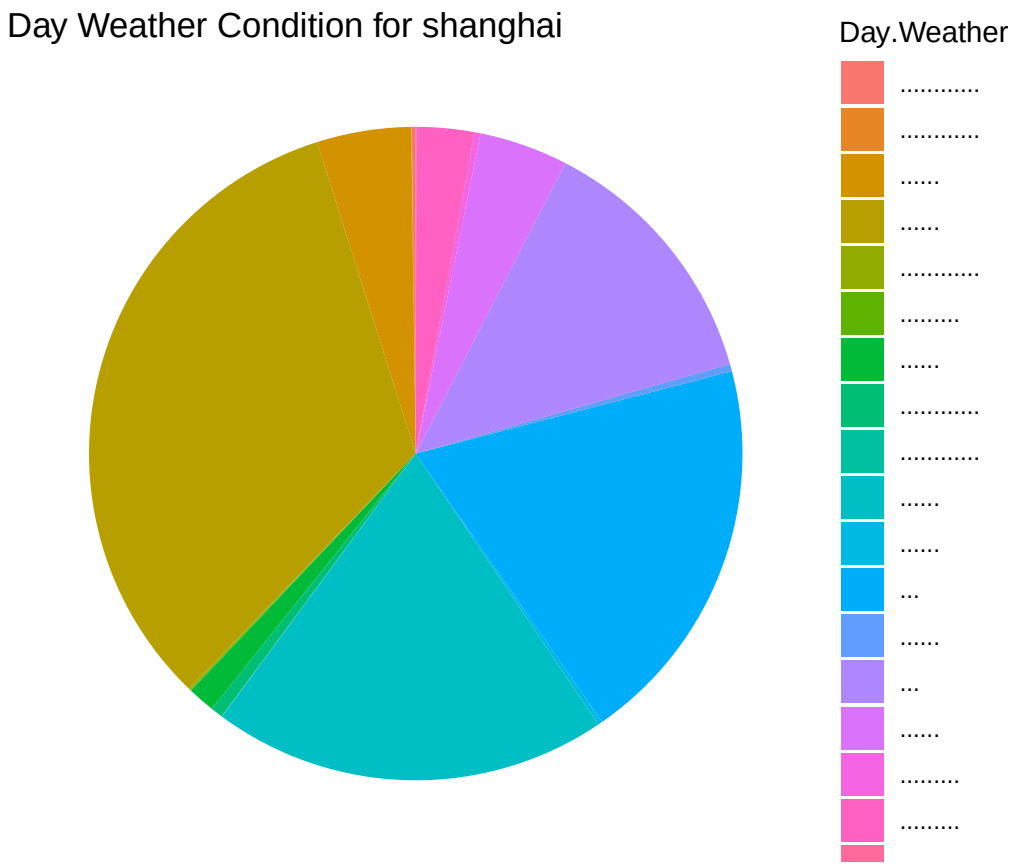

```r
day_weather_counts <- data %>%
  group_by(City, Day.Weather) %>%
  tally() %>%
  ungroup()

night_weather_counts <- data %>%
  group_by(City, Night.Weather) %>%
  tally() %>%
  ungroup()


# Function to plot pie chart for weather conditions by city
plot_pie_chart <- function(data, city, weather_column, title) {
  city_data <- data %>% filter(City == city)
  p <- ggplot(city_data, aes(x = "", y = n, fill = !!sym(weather_column))) +
    geom_bar(stat = "identity", width = 1) +
    coord_polar(theta = "y") +
    labs(title = paste(title, "for", city), fill = weather_column) +
    theme_void()
  return(p)
}

# Example: Plot pie chart for the first city in the dataset (you can loop for all cities)
cities <- unique(data$City)

# Create a list of plots for all cities
```

```r
plots <- list()

# Loop through cities to plot pie charts for Day and Night weather conditions

for (city in cities) {
  # Day Weather Pie Chart
  day_plot <- plot_pie_chart(day_weather_counts, city, "Day.Weather", "Day Weather Condition")
  print(day_plot)

  # Night Weather Pie Chart
  night_plot <- plot_pie_chart(night_weather_counts, city, "Night.Weather", "Night Weather Condition")
  print(night_plot)
}
```

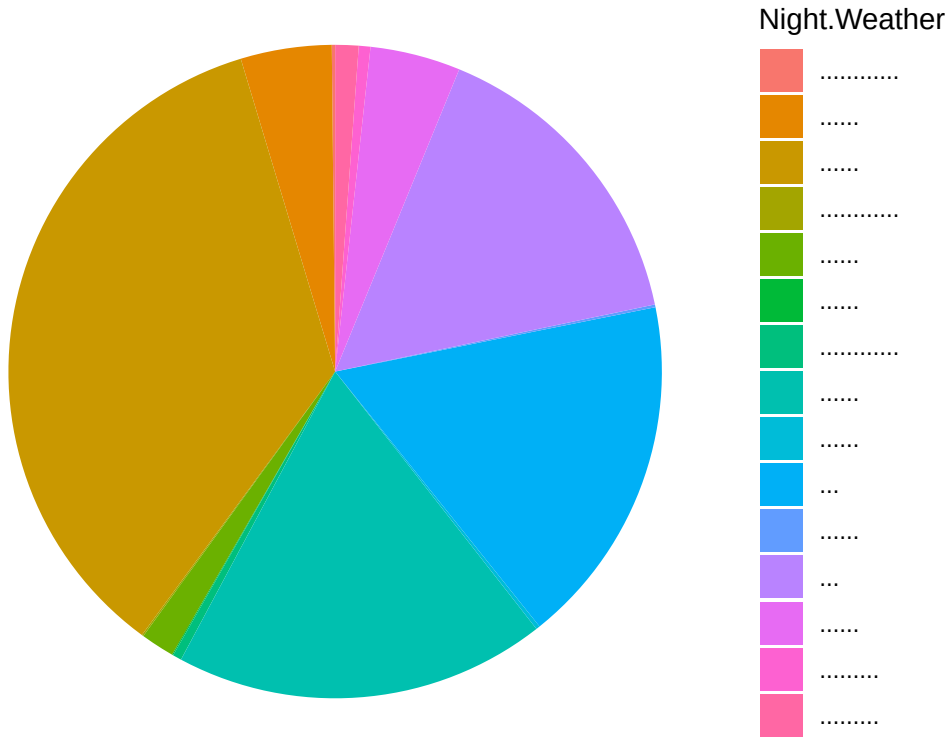


Night Weather Condition for beijing

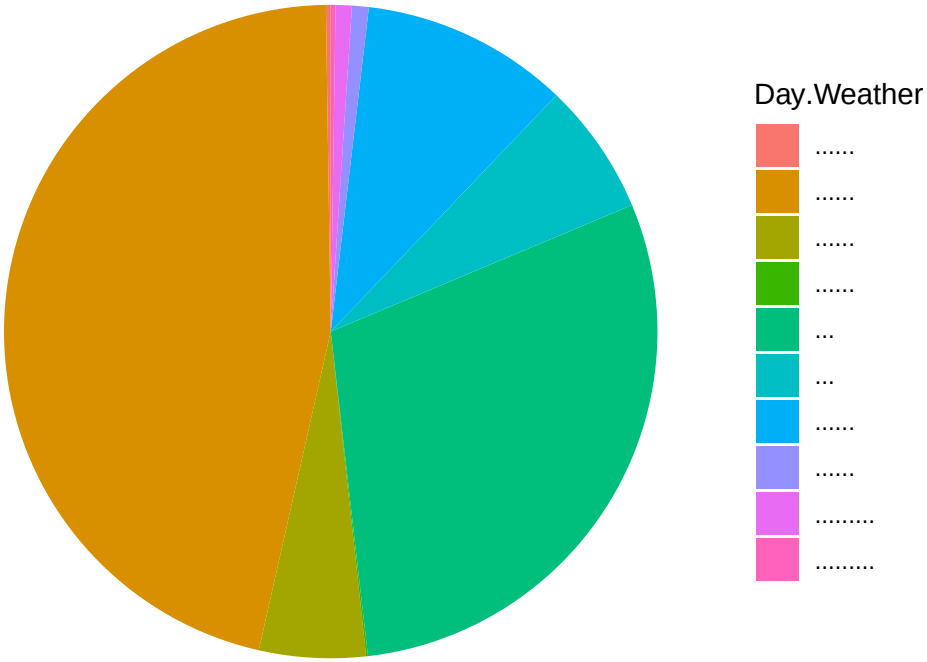



Day Weather Condition for shanghai
Day.Weather
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
...
.....
...
.....
.....
.....

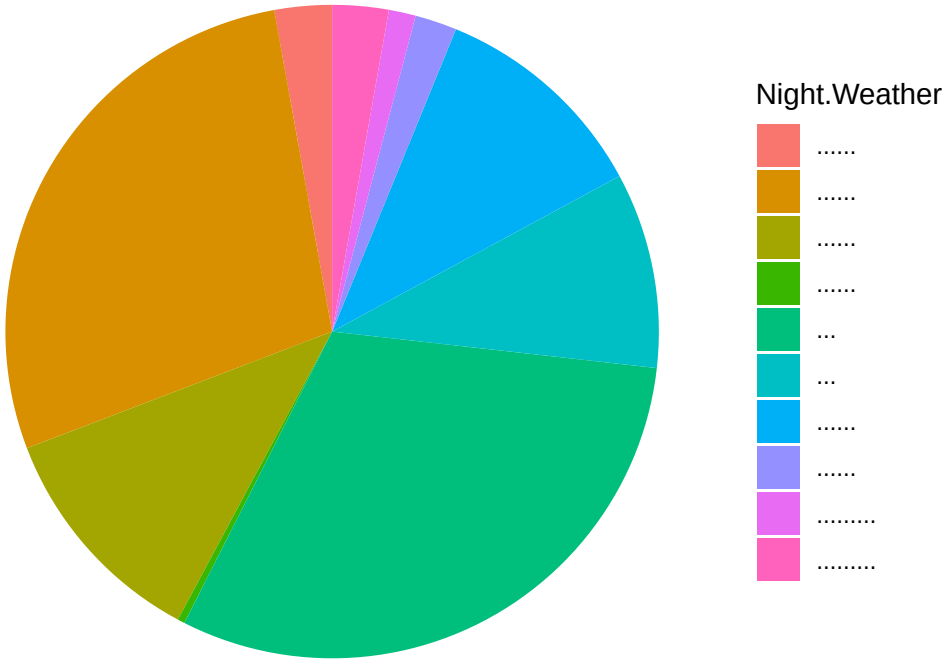
Night Weather Condition for shanghai



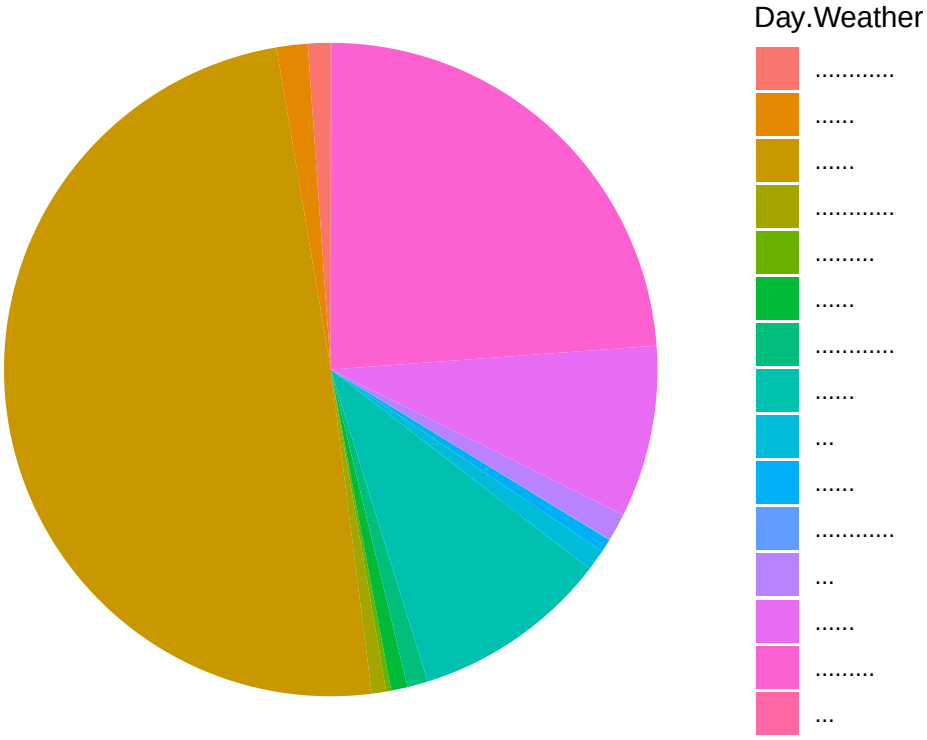
Day Weather Condition for lasa

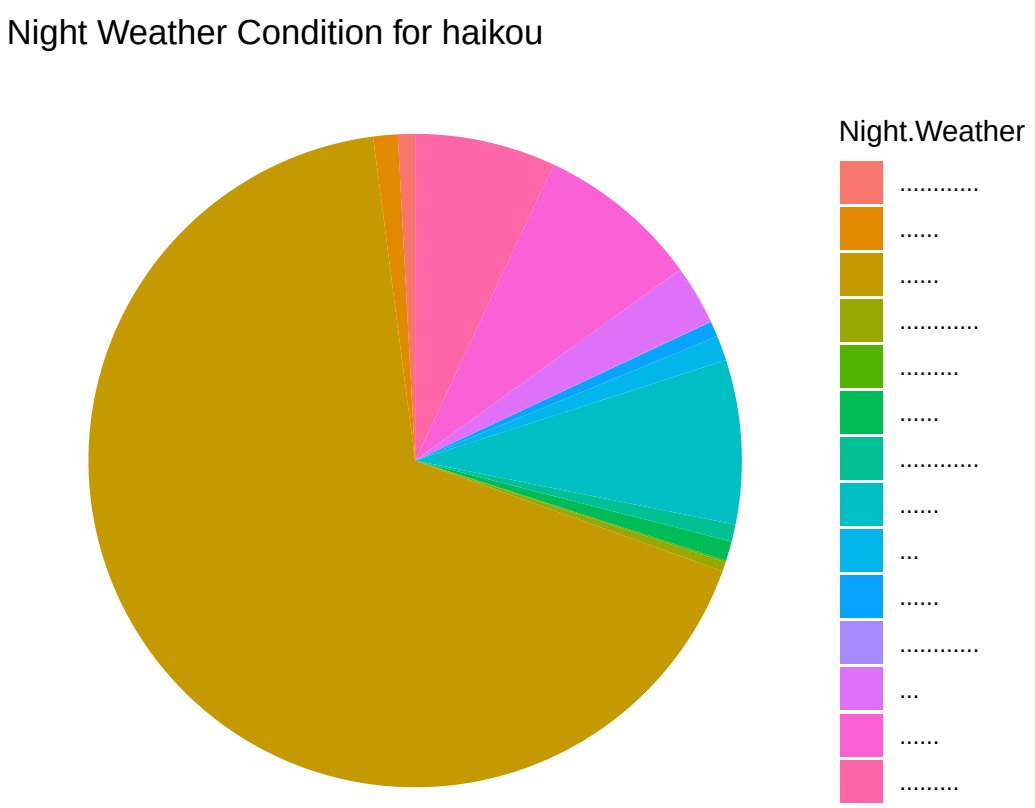


Night Weather Condition for lasa

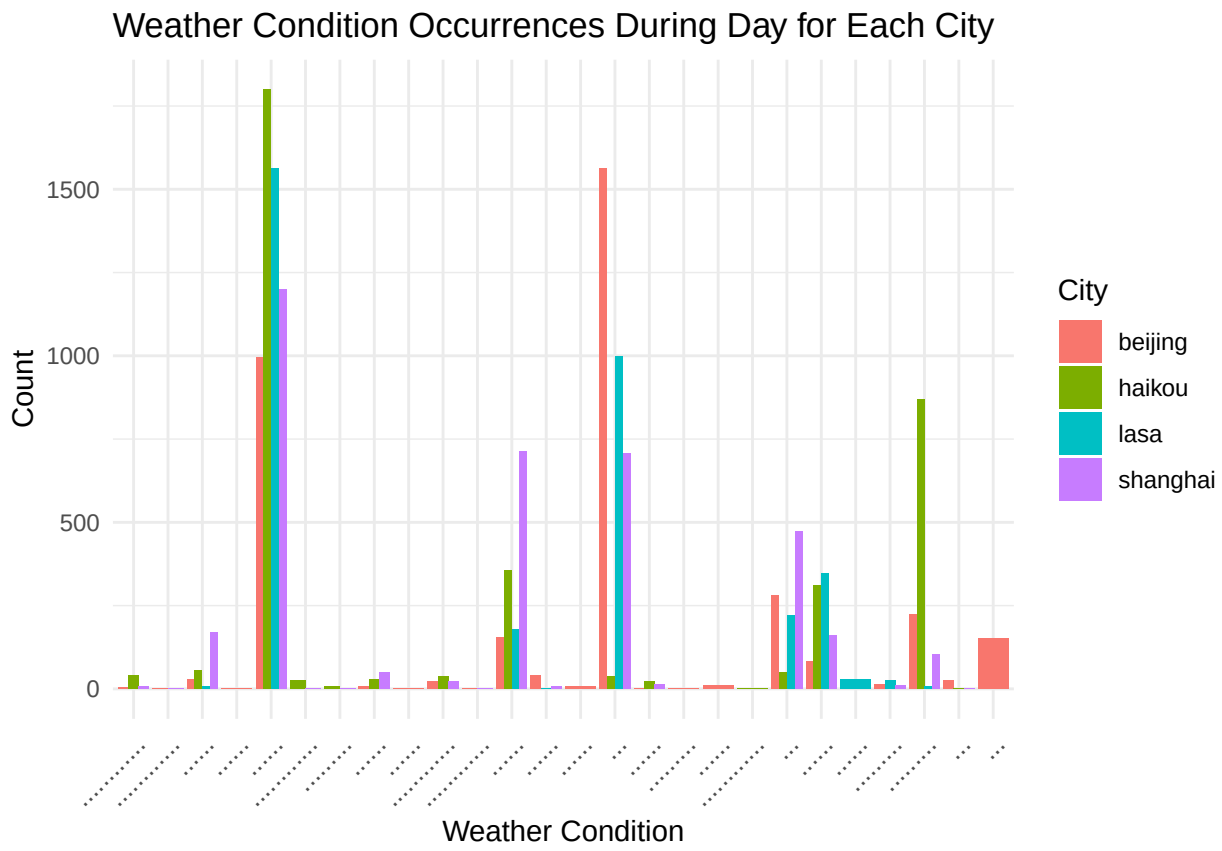


Day Weather Condition for haikou

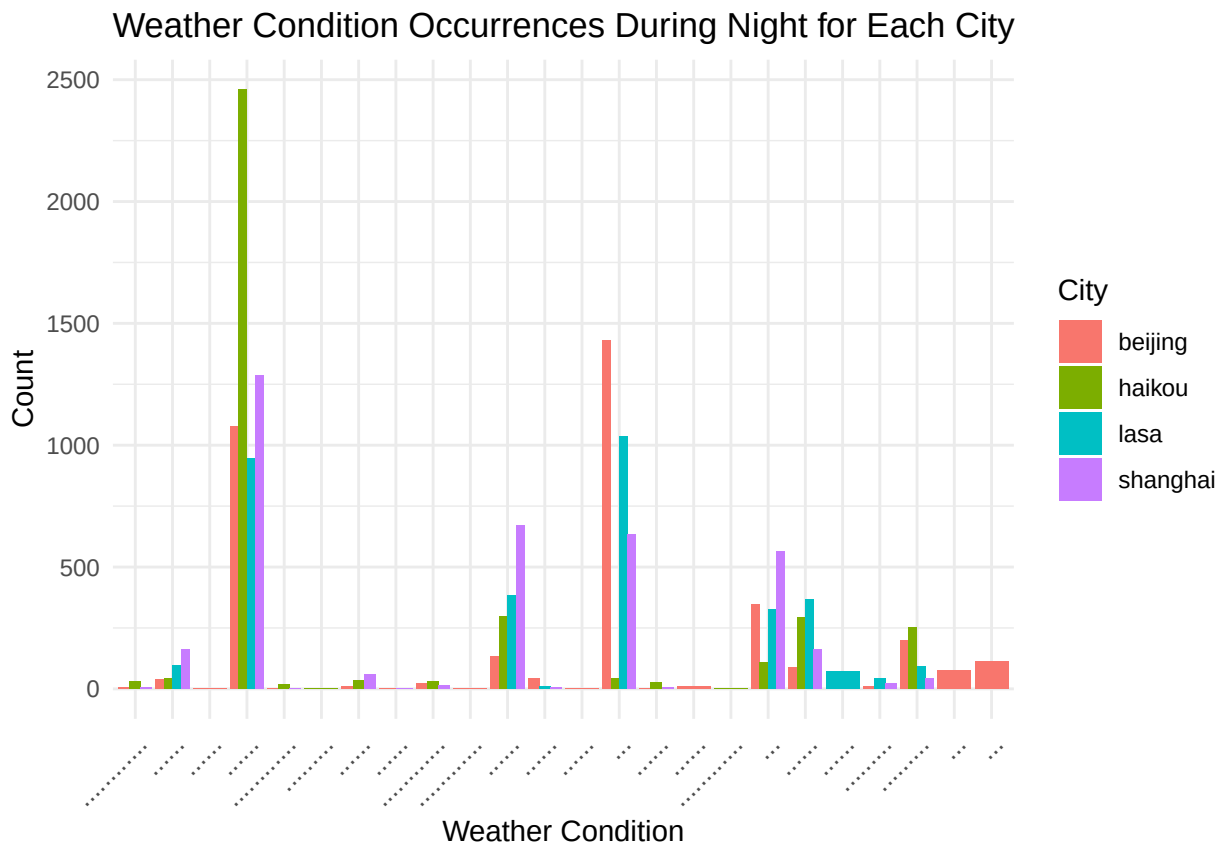





```
# Count the occurrences of each weather condition (Day and Night) by city
weather_condition_day <- data %>%
  group_by(City, Day.Weather) %>%
  summarise(Count = n())

weather_condition_night <- data %>%
  group_by(City, Night.Weather) %>%
  summarise(Count = n())

# Plot weather condition occurrences (day)
ggplot(weather_condition_day, aes(x = Day.Weather, y = Count, fill = City)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Weather Condition Occurrences During Day for Each City", x = "Weather Condition", y = "
  theme_minimal()+
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1, size = 10) # Rotate labels by 45 degrees
  )
```

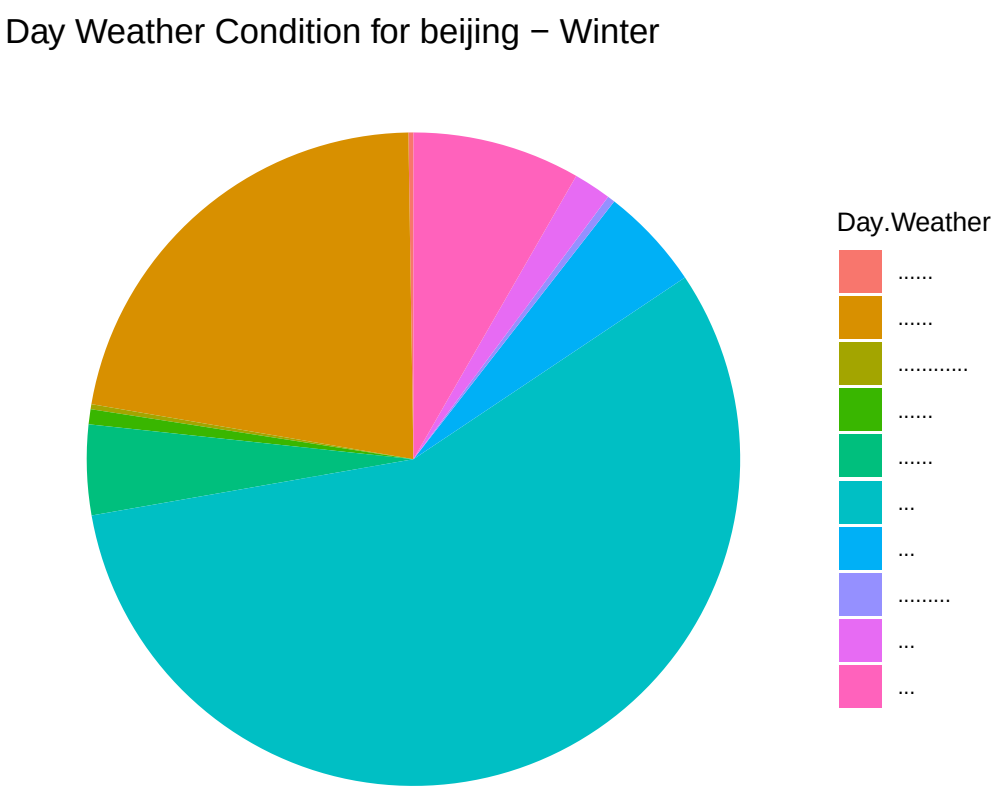


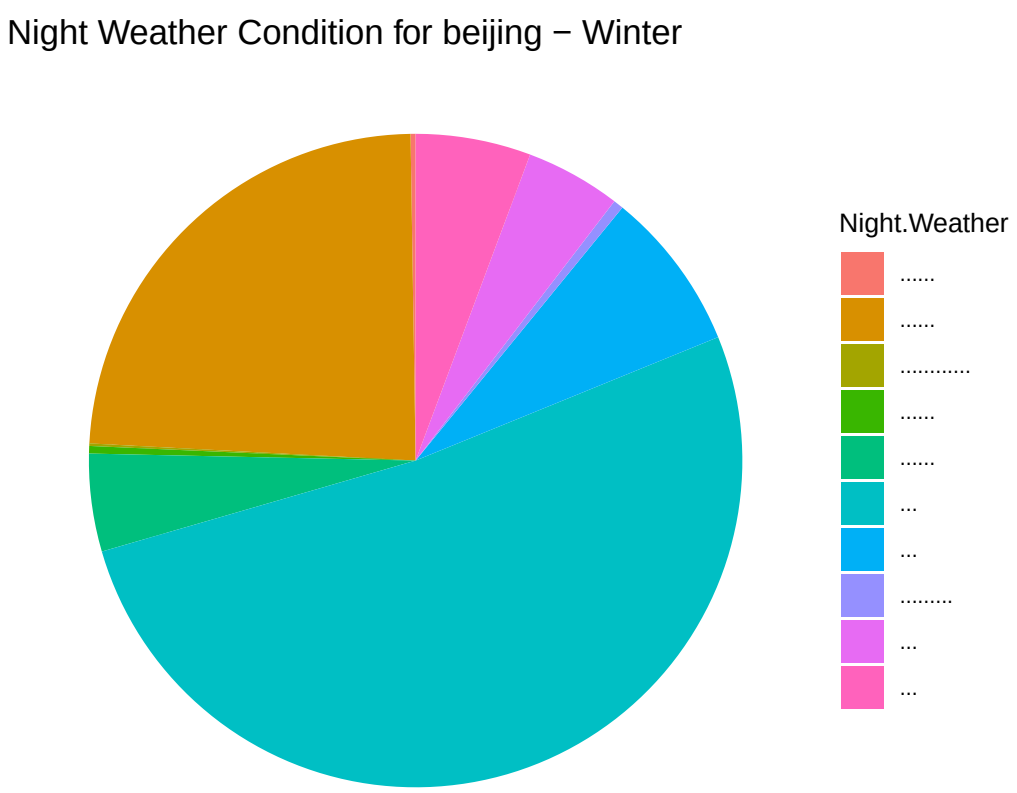

```
ggplot(weather_condition_night, aes(x = Night.Weather, y = Count, fill = City)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Weather Condition Occurrences During Night for Each City", x = "Weather Condition", y =
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1, size = 10) # Rotate labels by 45 degrees
  )
```

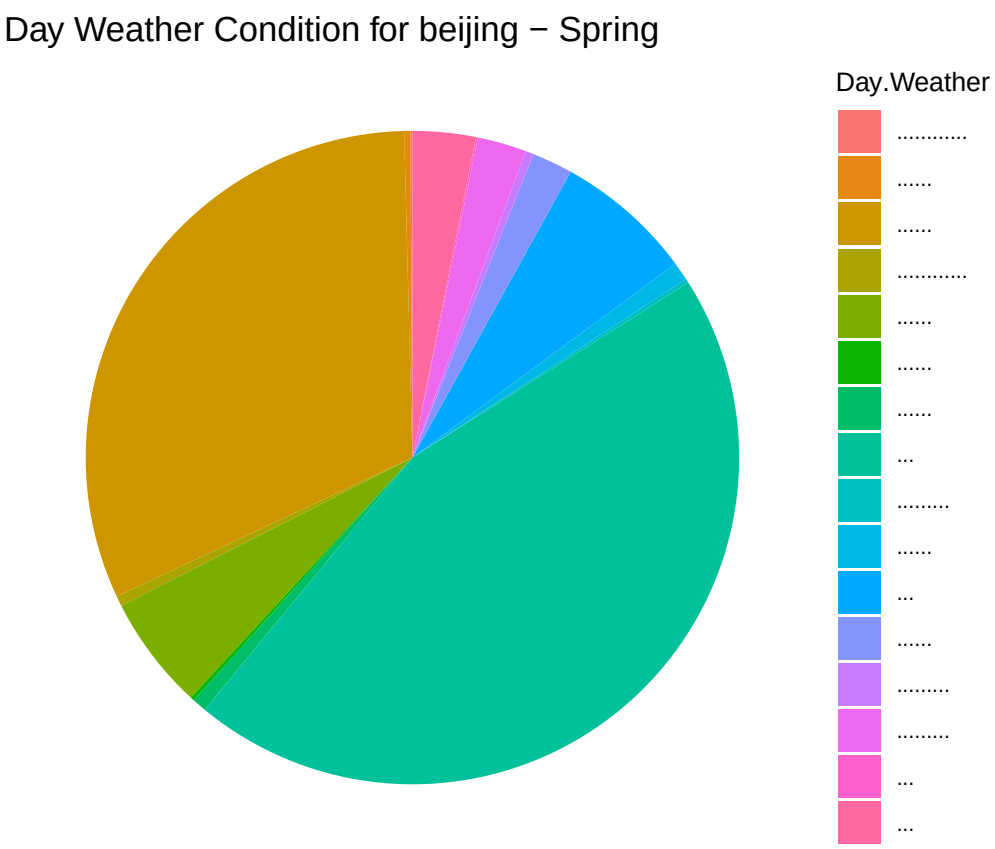


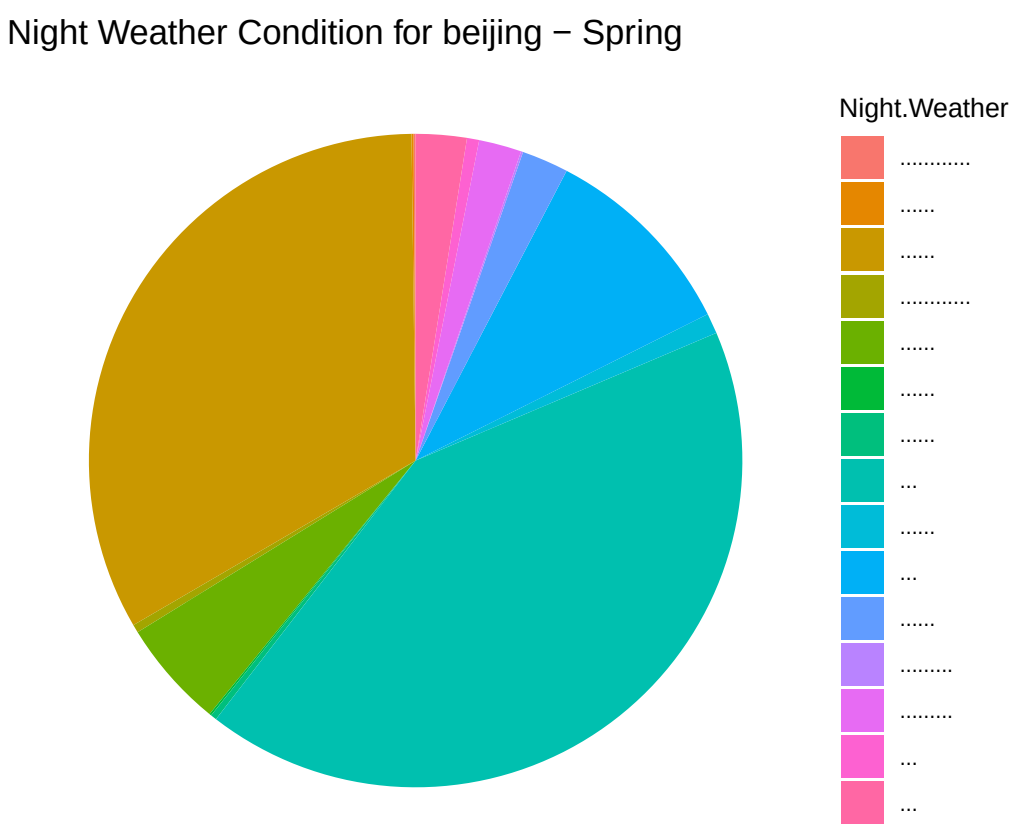

```
day_weather_counts <- data %>%
  group_by(City, Season, Day.Weather) %>%
  tally() %>%
  ungroup()

night_weather_counts <- data %>%
  group_by(City, Season, Night.Weather) %>%
  tally() %>%
  ungroup()

# Function to plot pie chart for weather conditions by city and season
plot_pie_chart <- function(data, city, season, weather_column, title) {
  city_season_data <- data %>% filter(City == city & Season == season)
  p <- ggplot(city_season_data, aes(x = "", y = n, fill = !!sym(weather_column))) +
    geom_bar(stat = "identity", width = 1) +
    coord_polar(theta = "y") +
    labs(title = paste(title, "for", city, "-", season), fill = weather_column) +
    theme_void() +
    theme(legend.position = "right",  # Move legend outside the plot
          legend.title = element_text(size = 10), # Adjust the legend title size
          legend.text = element_text(size = 8))   # Adjust the legend text size
  return(p)
}

# Loop through cities and seasons to plot pie charts for Day and Night weather conditions
cities <- unique(data$City)
```

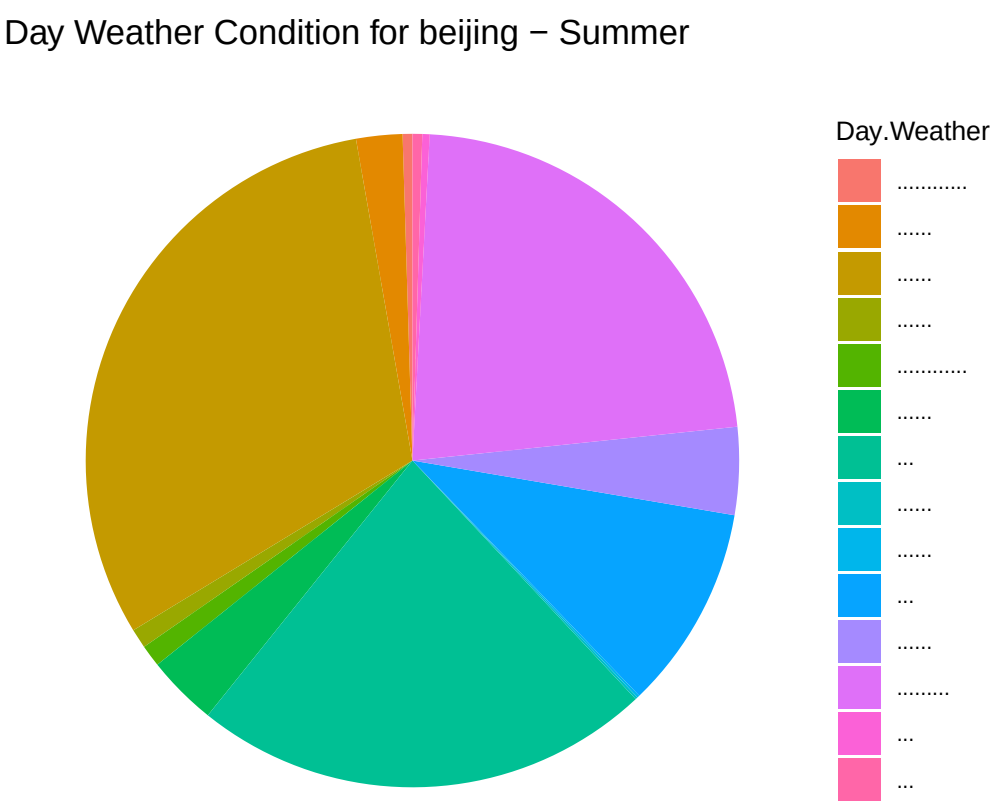
```r
seasons <- unique(data$Season)

for (city in cities) {
  for (season in seasons) {
    # Day Weather Pie Chart
    day_plot <- plot_pie_chart(day_weather_counts, city, season, "Day.Weather", "Day Weather Condition")
    print(day_plot)

    # Night Weather Pie Chart
    night_plot <- plot_pie_chart(night_weather_counts, city, season, "Night.Weather", "Night Weather Co
    print(night_plot)
  }
}
```



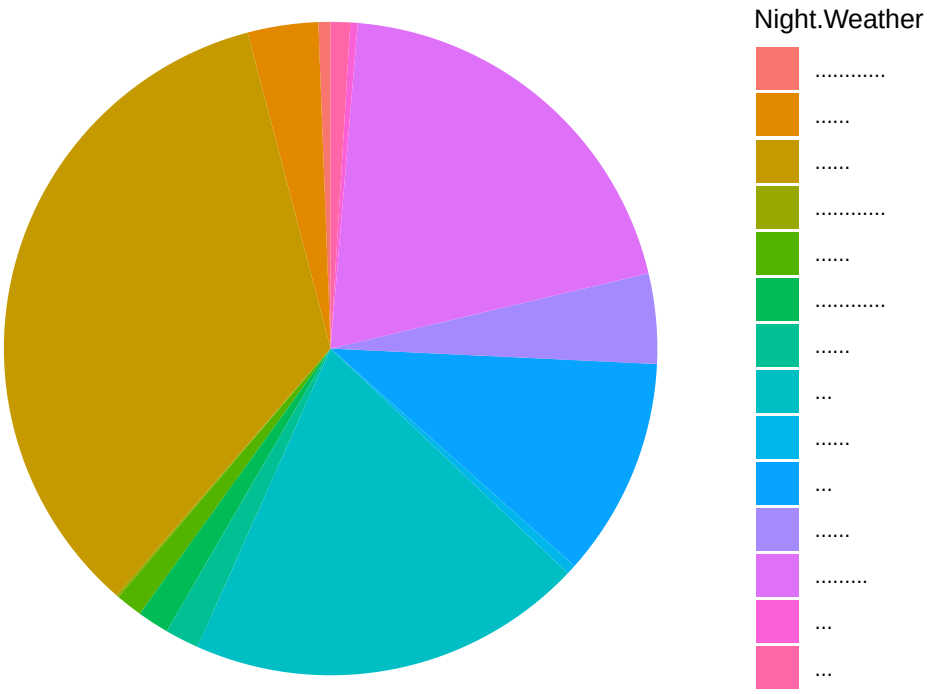

Night Weather Condition for beijing − Winter
Night.Weather
.....
.....
.....
.....
.....
...
...
.....
...
...

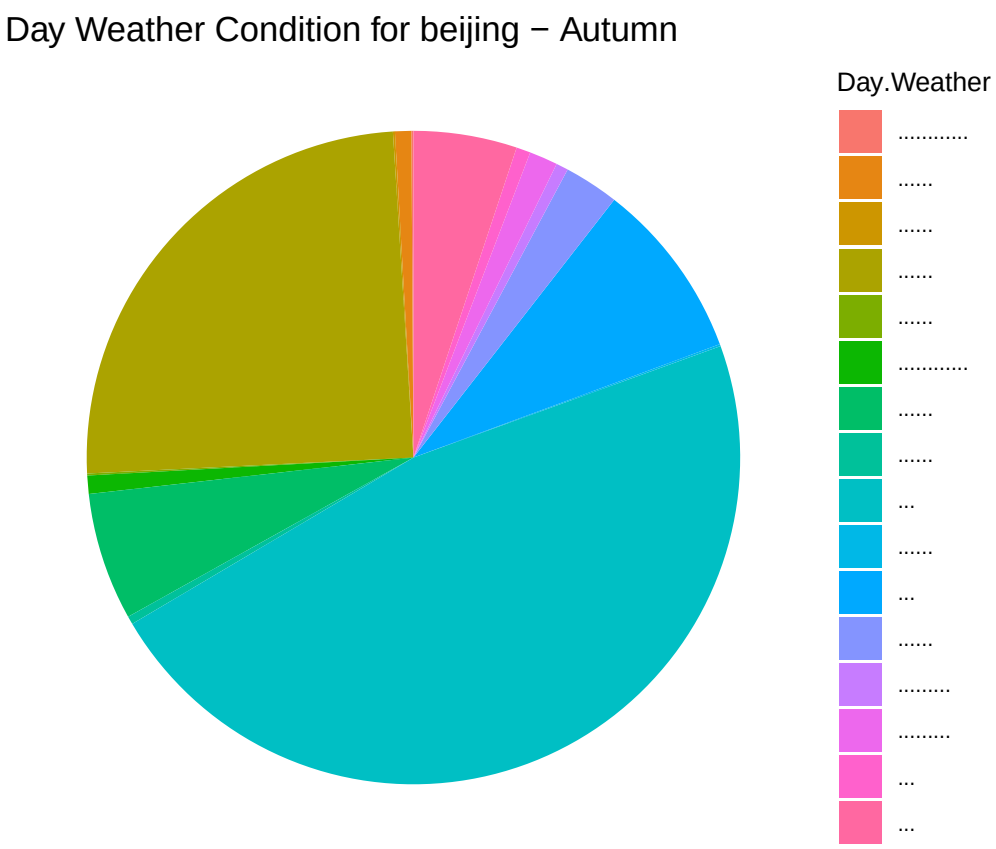

Day Weather Condition for beijing − Spring
Day.Weather
.....
.....
.....
.....
.....
.....
.....
...
.....
.....
...
.....
.....
.....
...
...

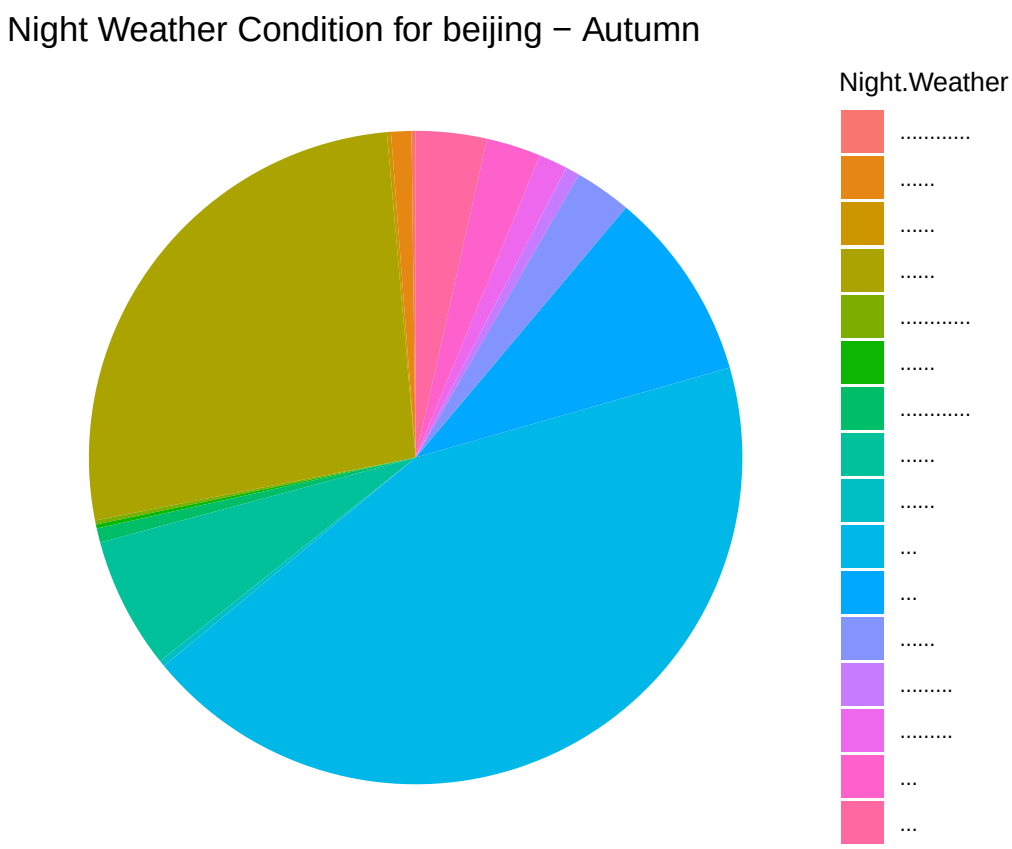

Night Weather Condition for beijing − Spring
Night.Weather
.....
.....
.....
.....
.....
.....
.....
...
.....
...
.....
.....
.....
...
...


Day Weather Condition for beijing − Summer
Day.Weather
.....
.....
.....
.....
.....
.....
...
.....
.....
...
.....
.....
...
...

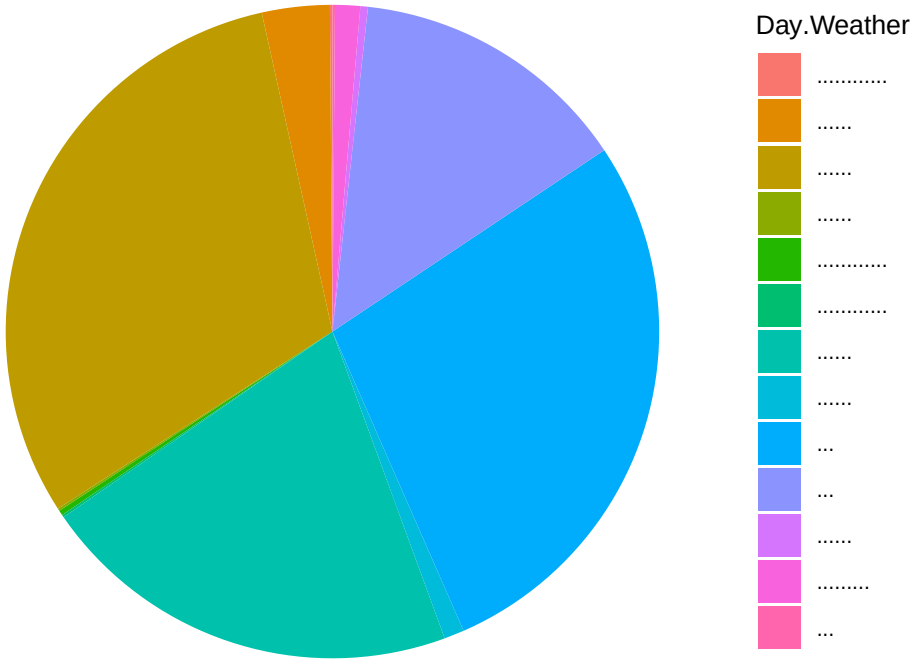
Night Weather Condition for beijing − Summer



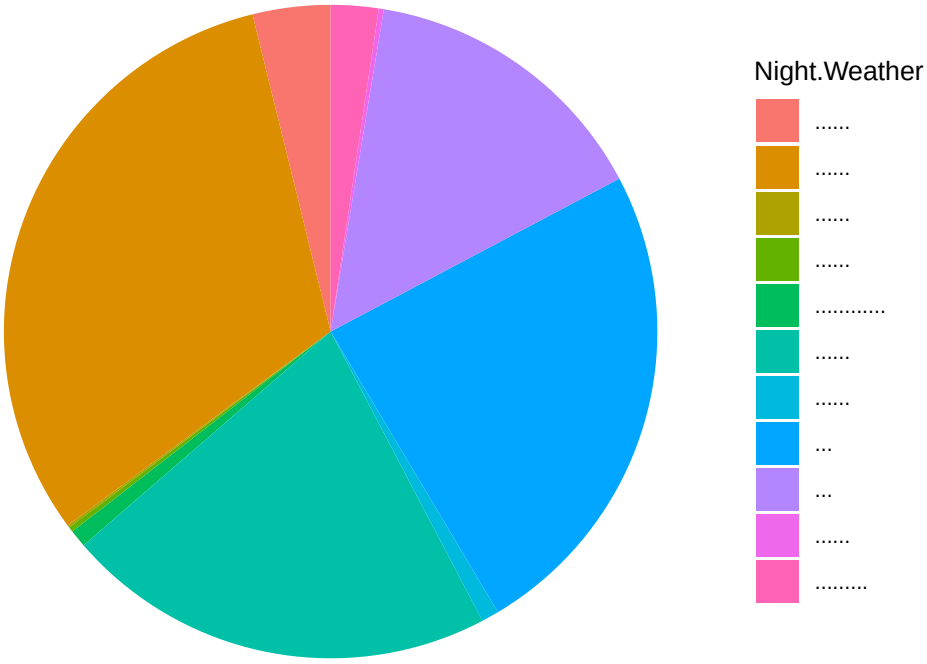

Day Weather Condition for beijing − Autumn
Day.Weather
.....
.....
.....
.....
.....
.....
.....
.....
...
.....
...
.....
.....
.....
...
...

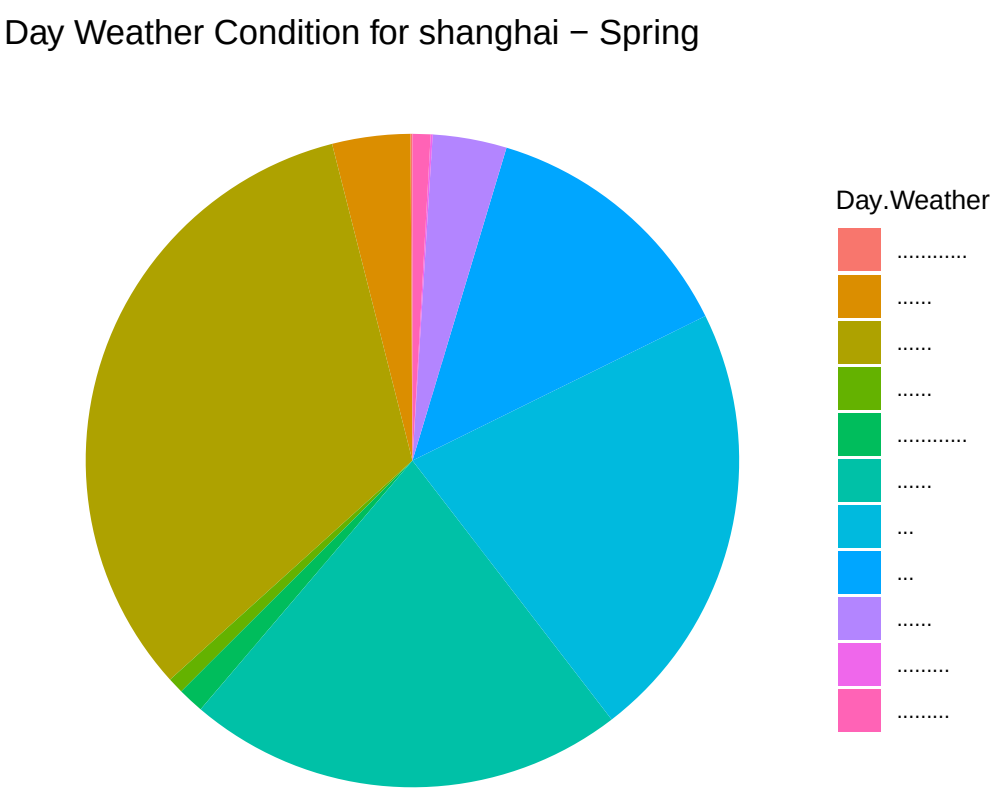

Night Weather Condition for beijing − Autumn
Night.Weather
.....
.....
.....
.....
.....
.....
.....
.....
.....
...
...
.....
.....
.....
...
...

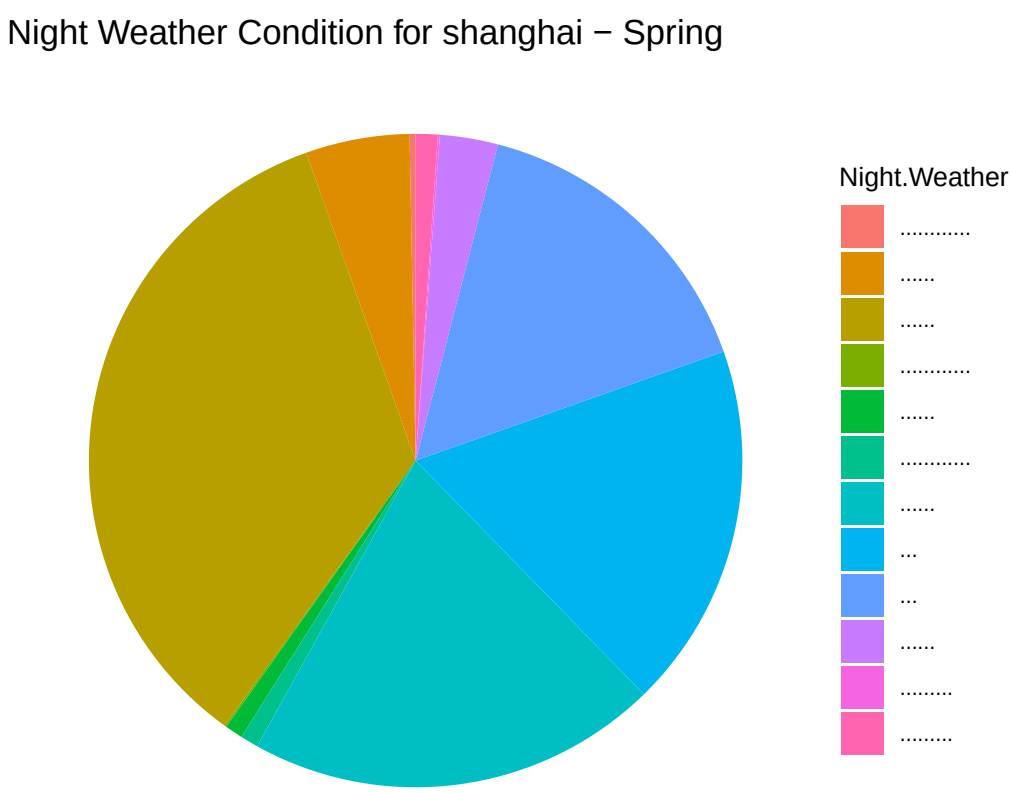
Day Weather Condition for shanghai − Winter

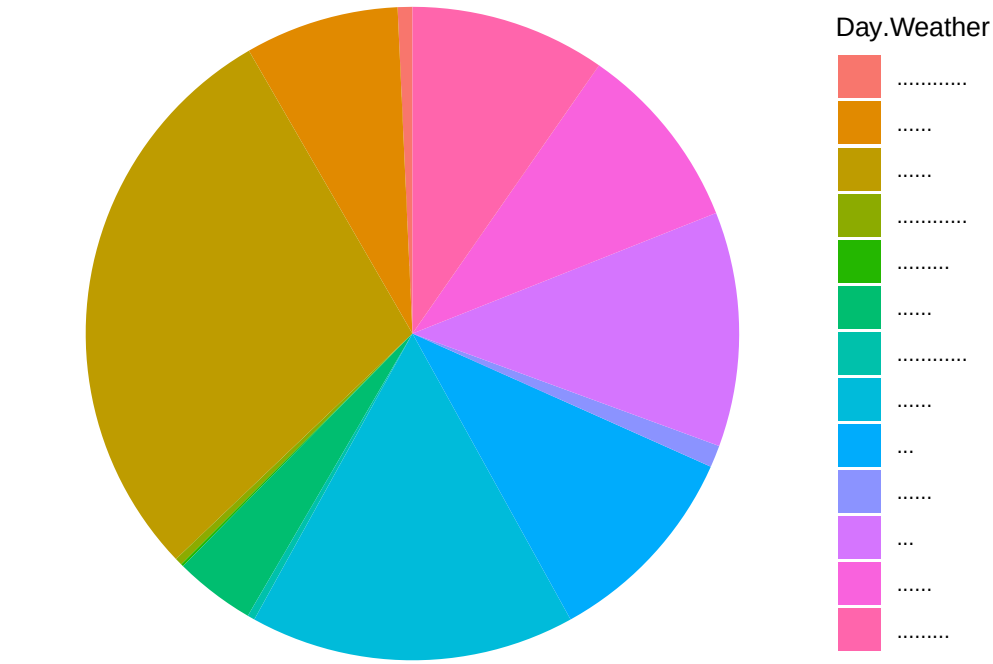


Night Weather Condition for shanghai − Winter

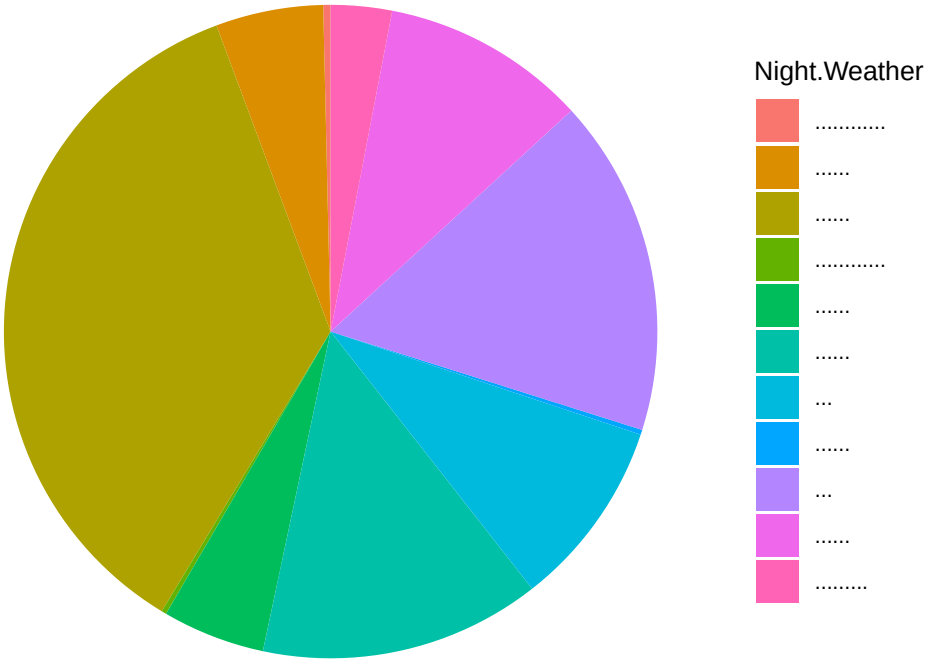


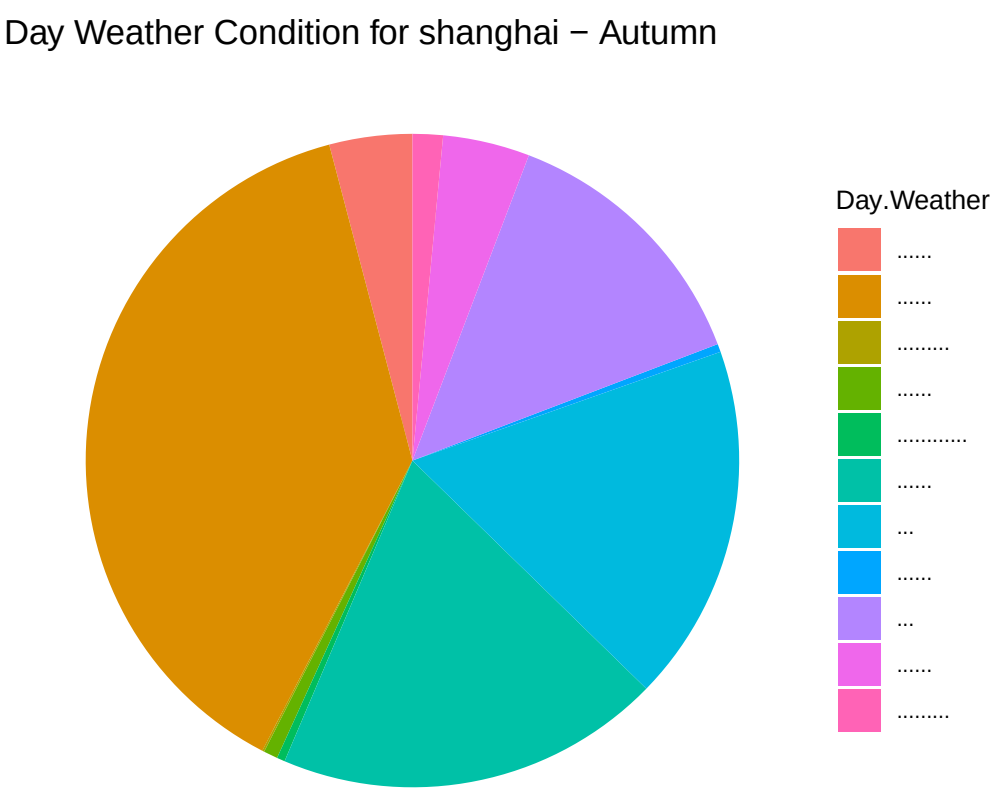

Day Weather Condition for shanghai − Spring
Day.Weather
.....
.....
.....
.....
.....
.....
...
...
.....
.....
.....

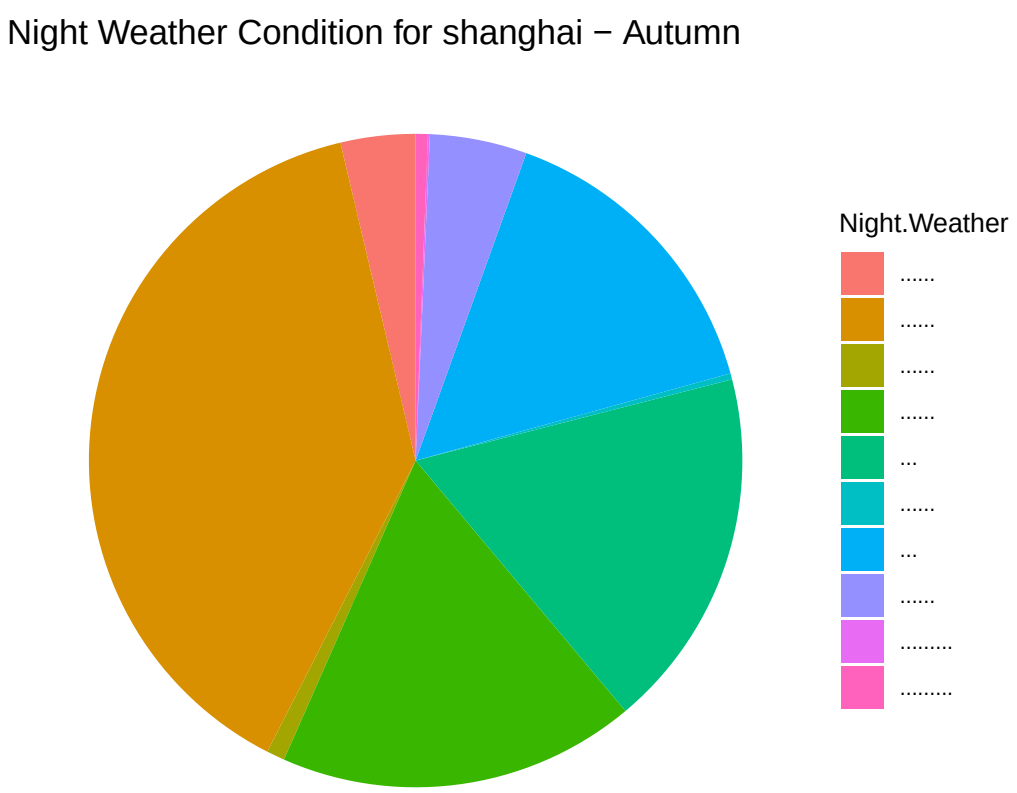

Night Weather Condition for shanghai − Spring
Night.Weather
.....
.....
.....
.....
.....
.....
.....
...
...
.....
.....
.....

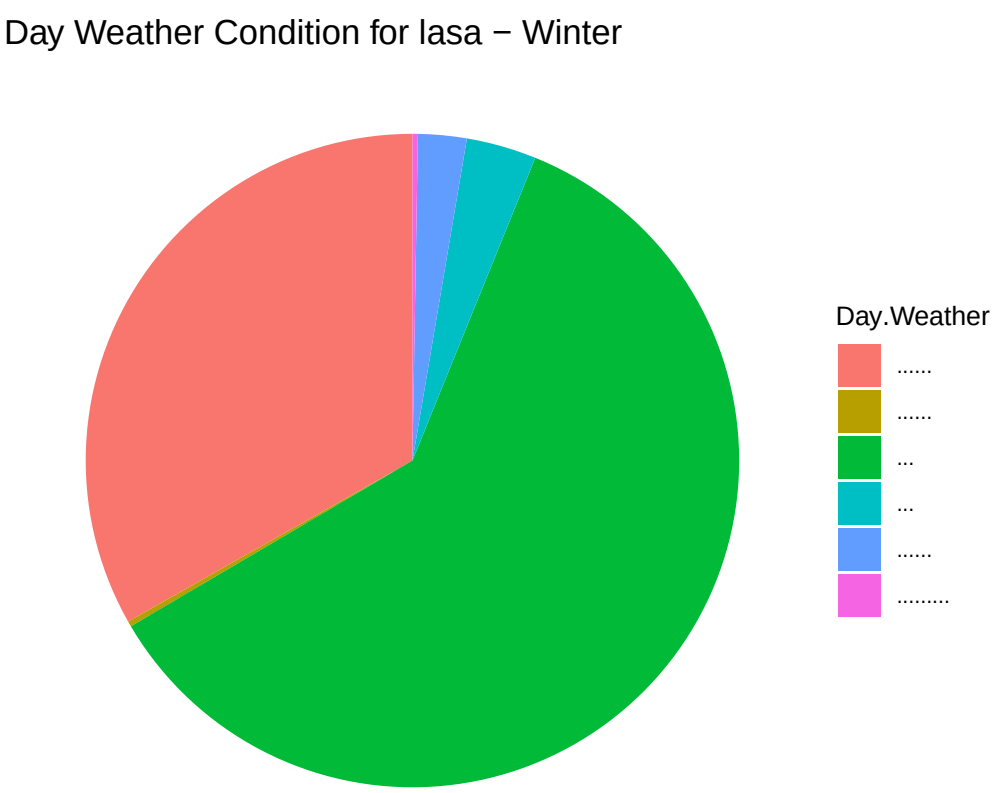

Day Weather Condition for shanghai − Summer
Day.Weather
.....
.....
.....
.....
.....
.....
.....
.....
...
.....
...
.....
.....

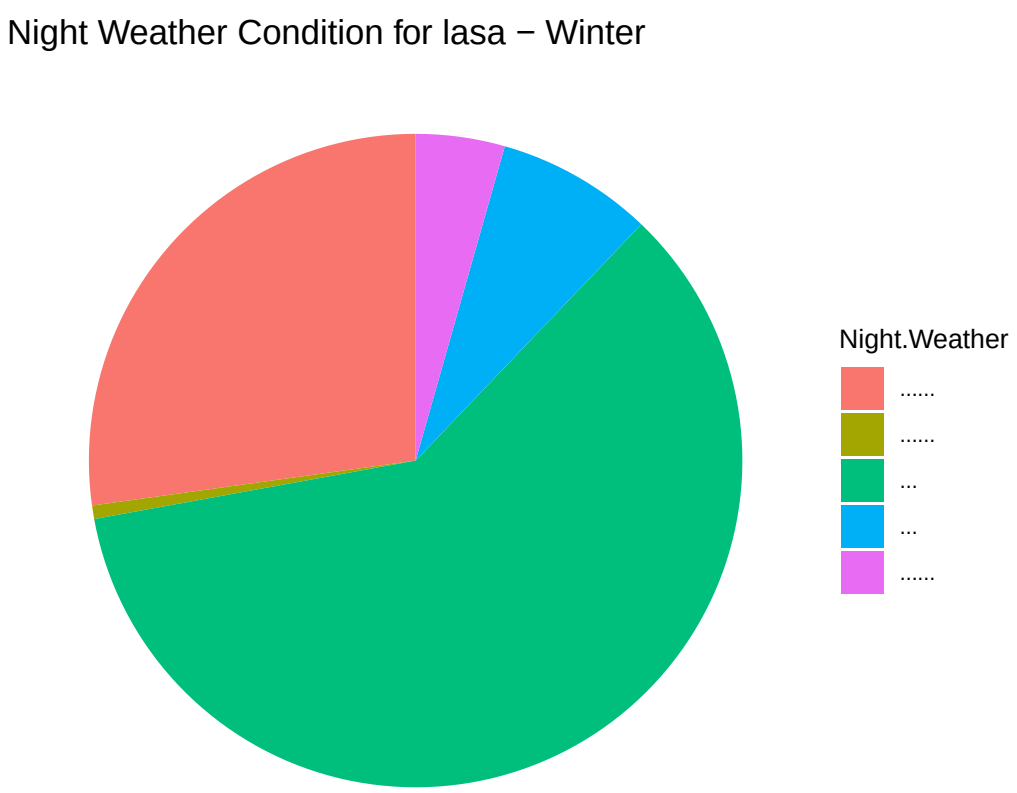
Night Weather Condition for shanghai − Summer

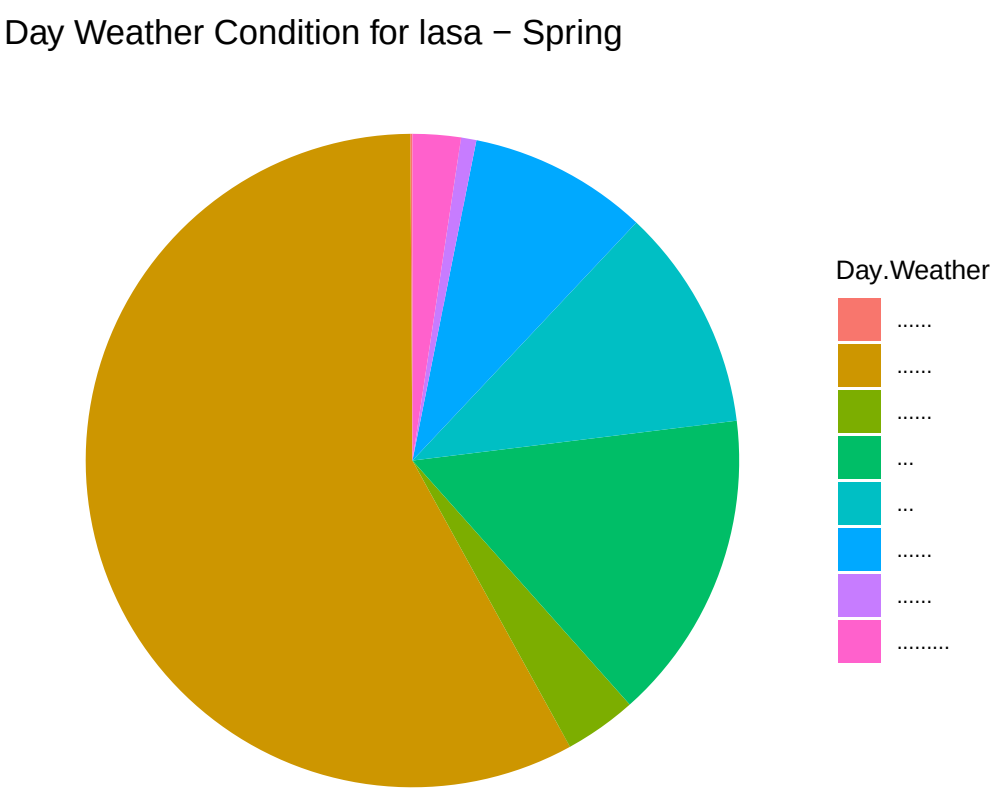


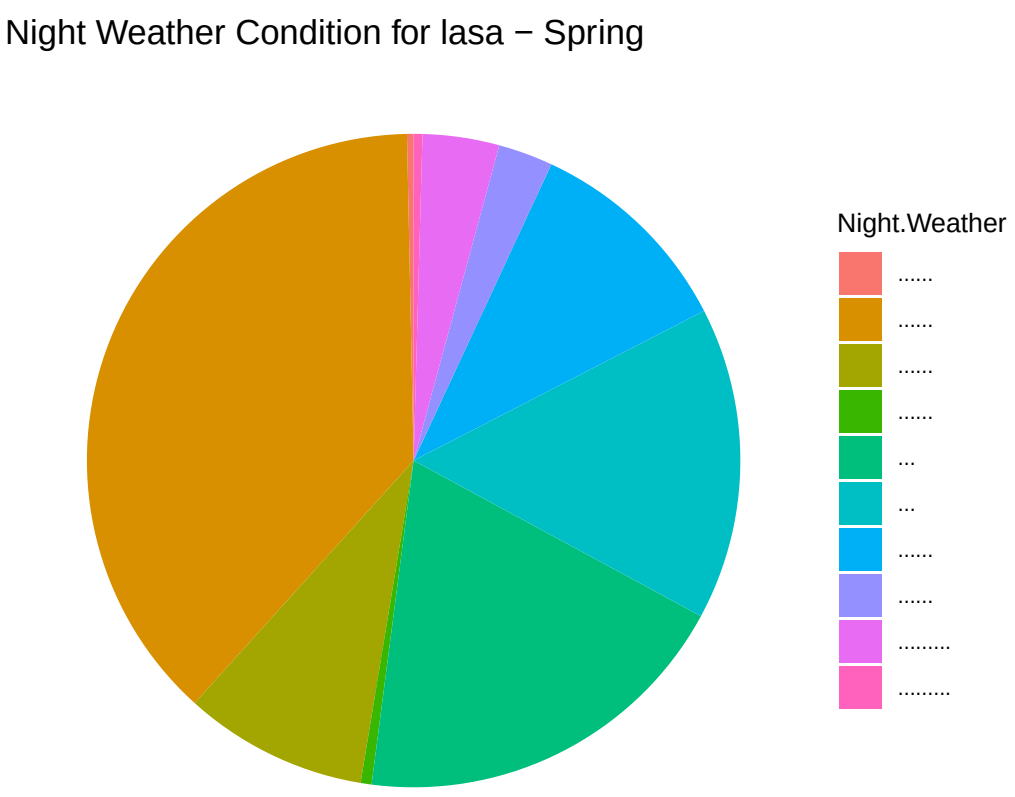

Day Weather Condition for shanghai − Autumn
Day.Weather
.....
.....
.....
.....
.....
.....
...
.....
...
.....
.....

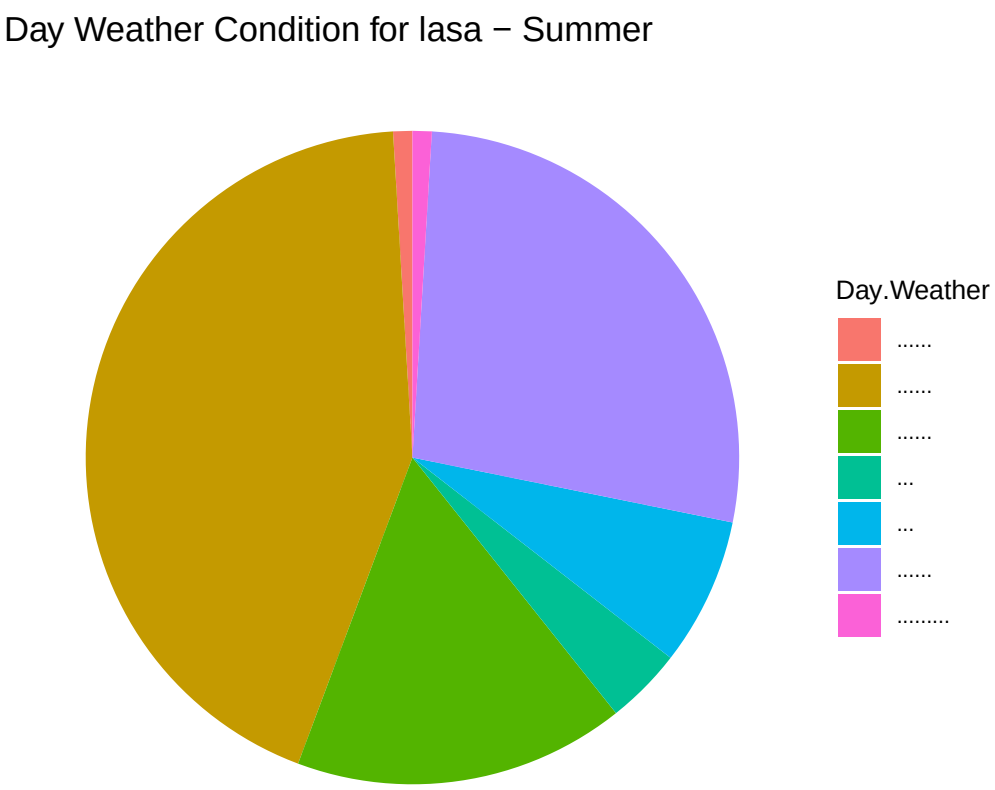

Night Weather Condition for shanghai − Autumn
Night.Weather
.....
.....
.....
.....
...
.....
...
.....
.....
.....

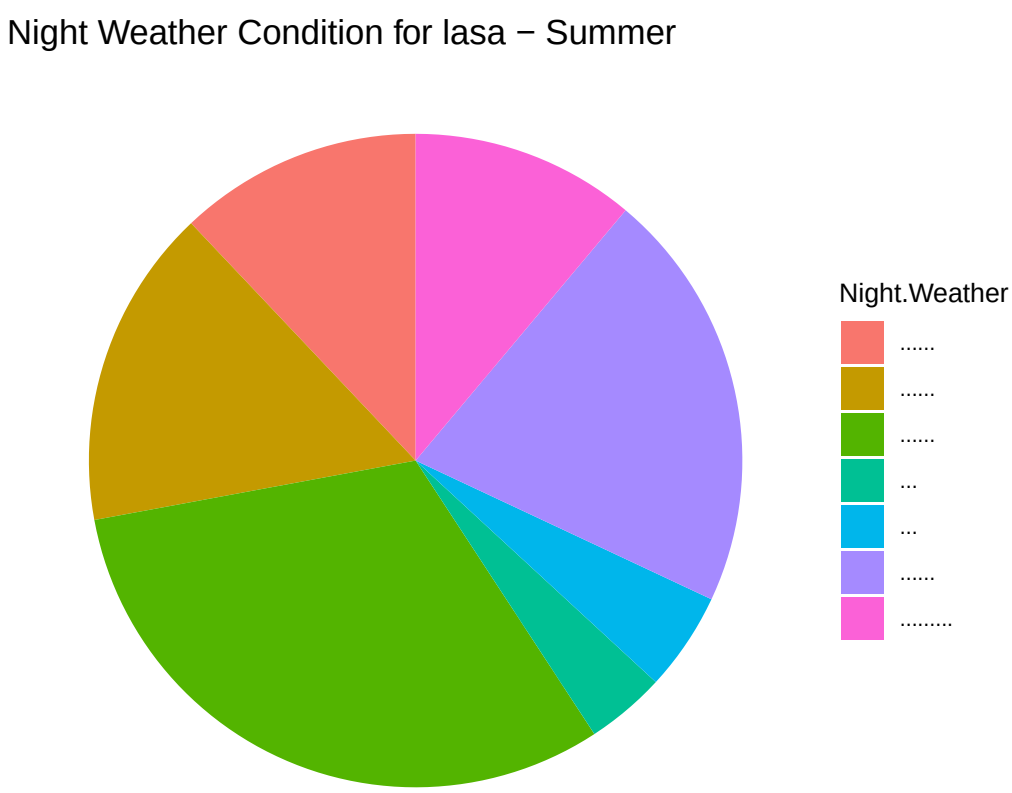

Day Weather Condition for lasa − Winter
Day.Weather
.....
.....
...
...
.....
.....

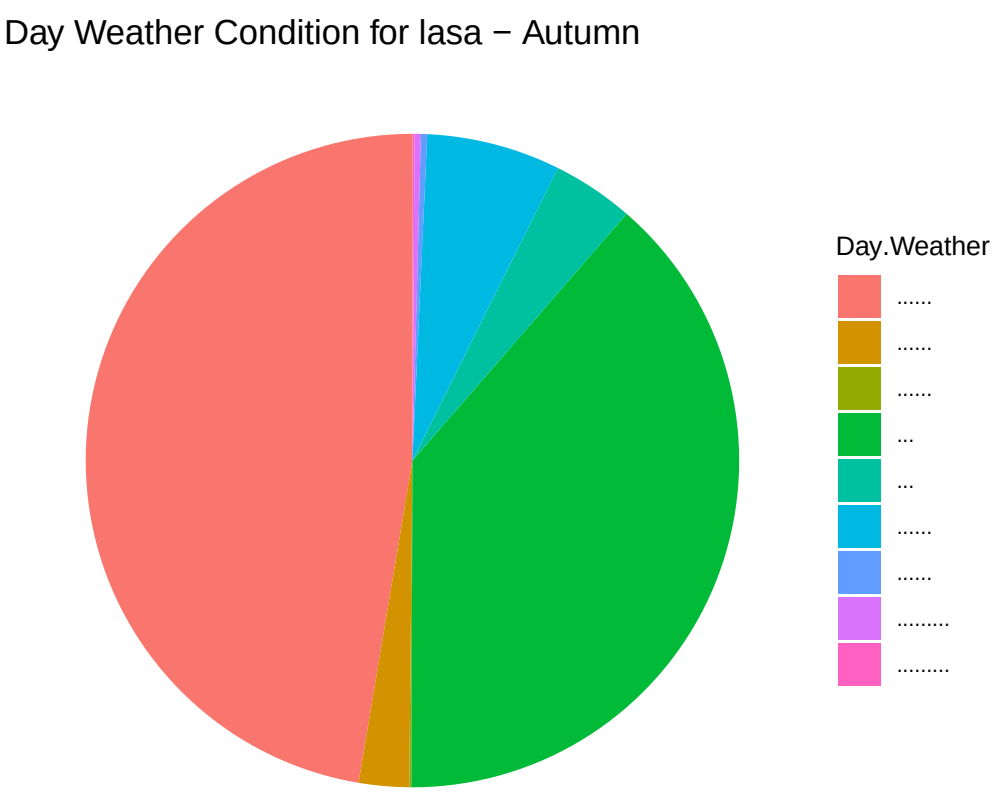

Night Weather Condition for lasa − Winter
Night.Weather
.....
.....
...
...
.....


Day Weather Condition for lasa − Spring
Day.Weather
.....
.....
.....
...
...
.....
.....
.....

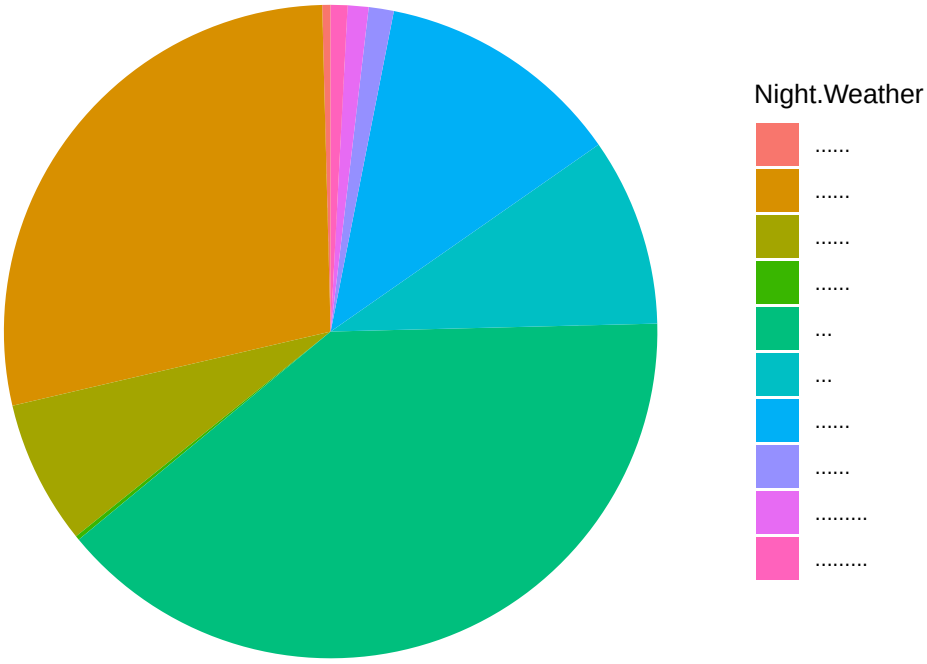

Night Weather Condition for lasa − Spring
Night.Weather
.....
.....
.....
.....
...
...
.....
.....
.....
.....

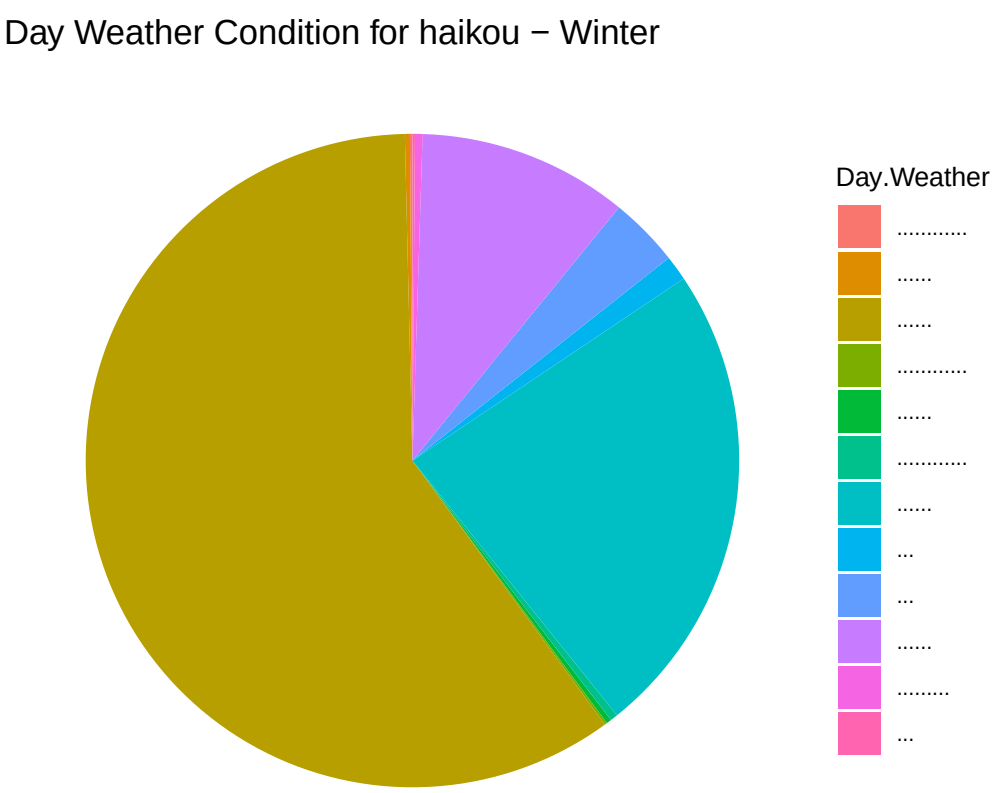

Day Weather Condition for lasa − Summer
Day.Weather
.....
.....
.....
...
...
.....
.....

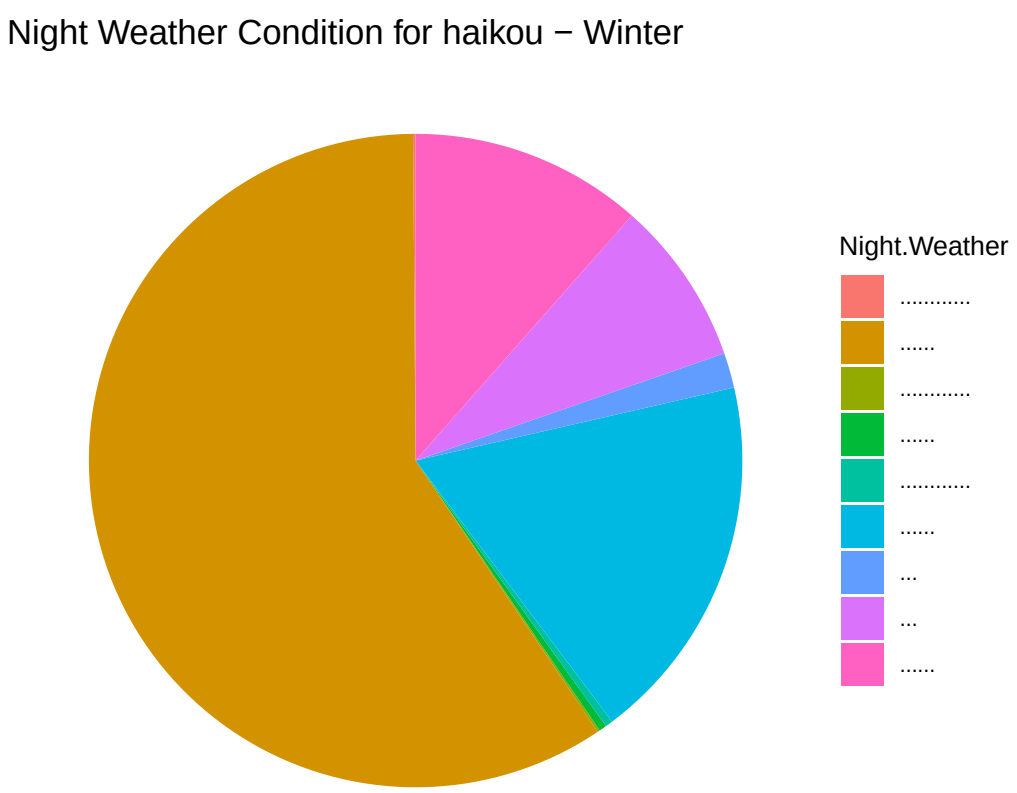

Night Weather Condition for lasa − Summer
Night.Weather
.....
.....
.....
...
...
.....
.....


Day Weather Condition for lasa − Autumn
Day.Weather
.....
.....
.....
...
...
.....
.....
.....
.....

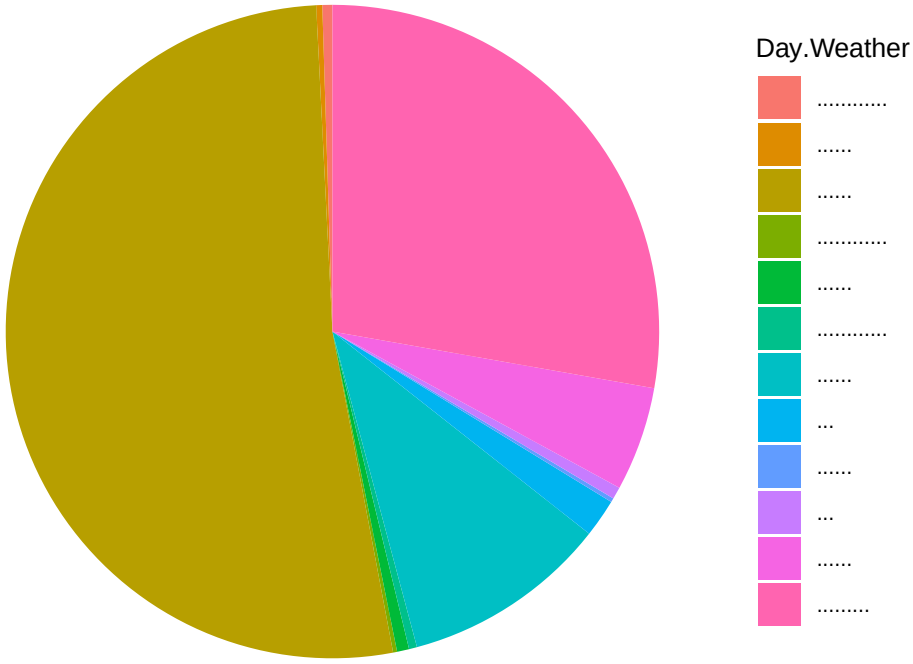
Night Weather Condition for lasa − Autumn

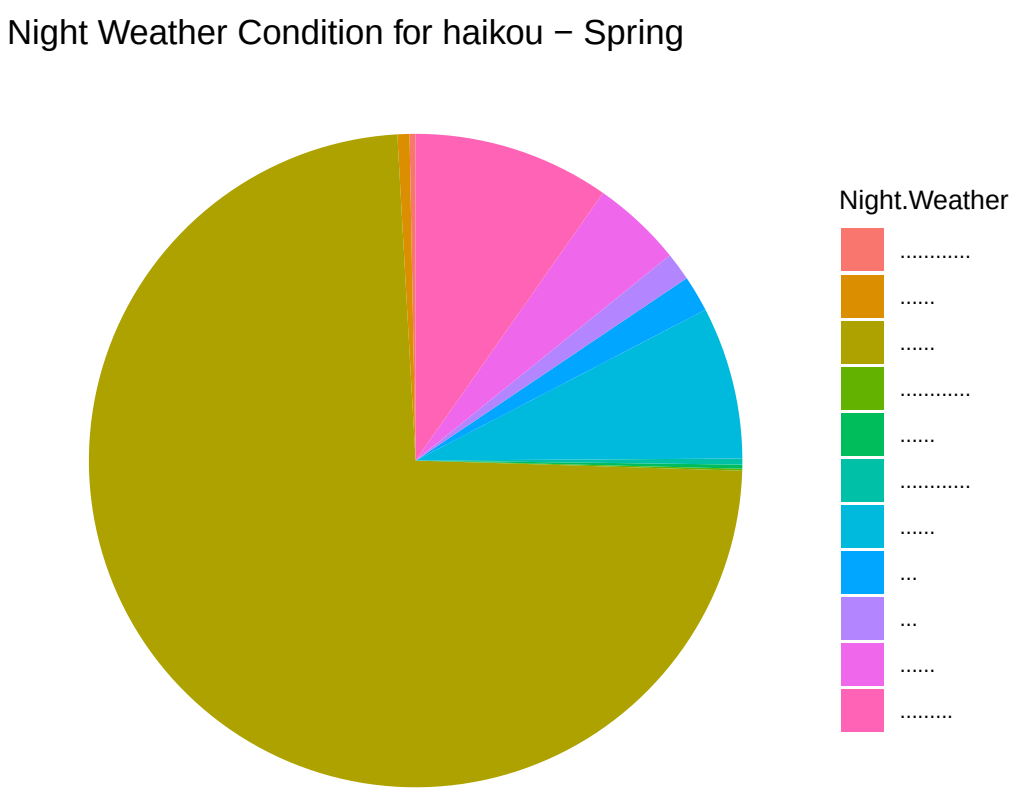


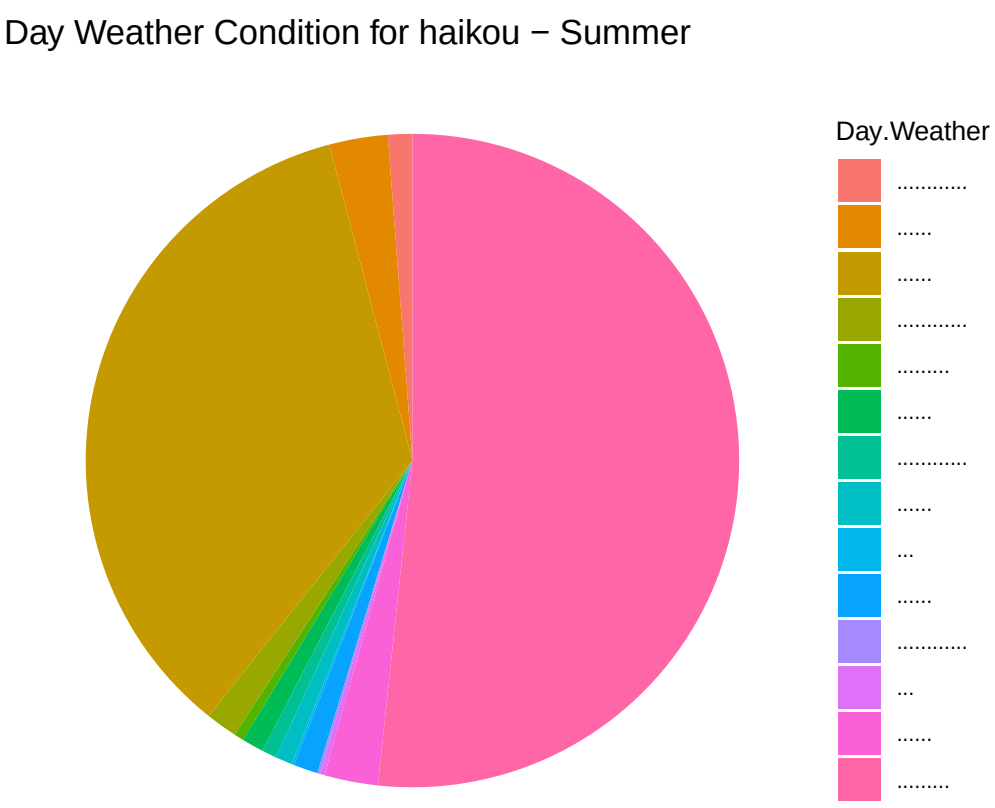

Day Weather Condition for haikou − Winter
Day.Weather
.....
.....
.....
.....
.....
.....
.....
...
...
.....
.....
...


Night Weather Condition for haikou − Winter
Night.Weather
.....
.....
.....
.....
.....
.....
...
...
.....

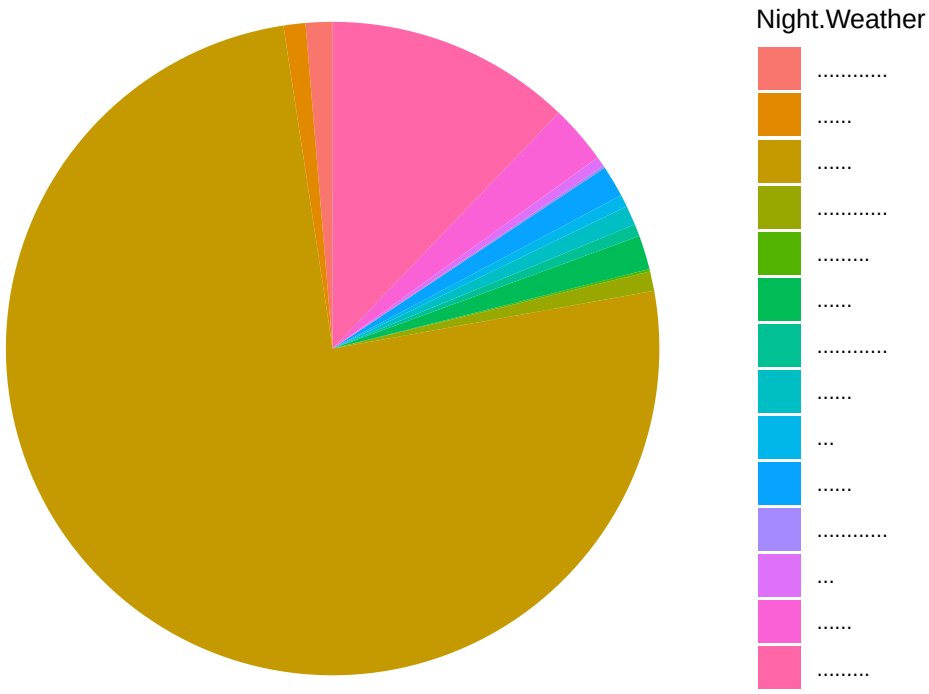
Day Weather Condition for haikou − Spring

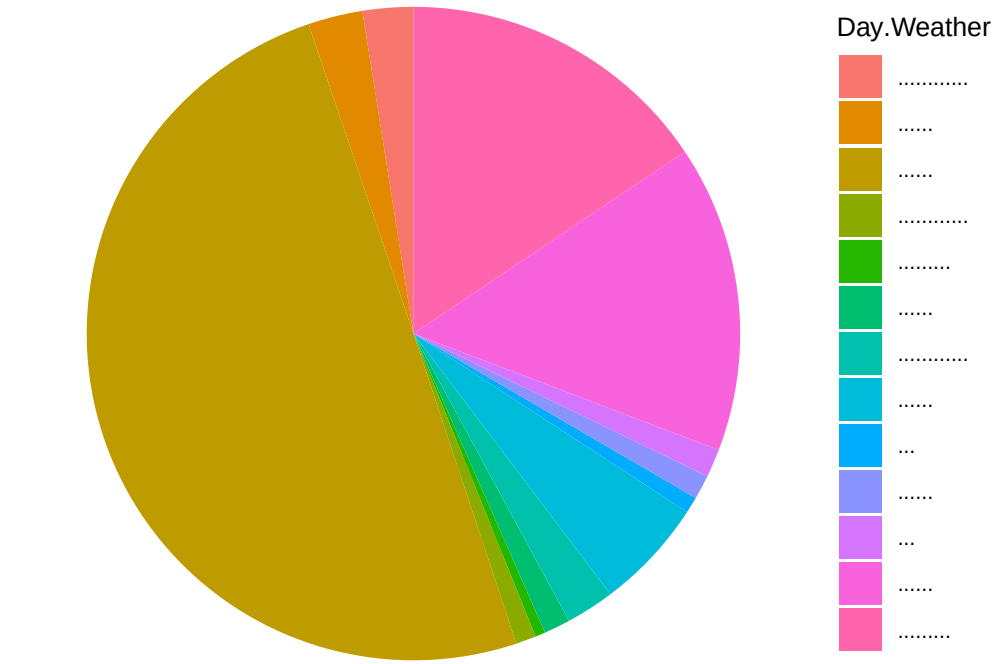


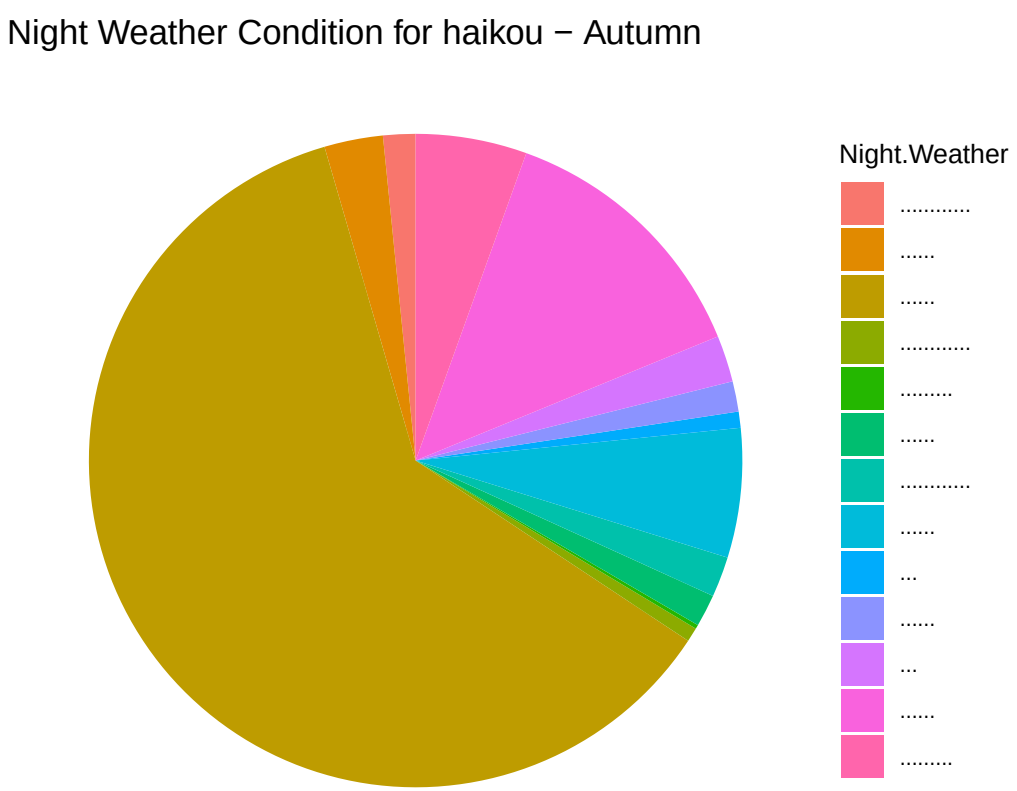

Night Weather Condition for haikou − Spring
Night.Weather
.....
.....
.....
.....
.....
.....
.....
...
...
.....
.....


Day Weather Condition for haikou − Summer
Day.Weather
.....
.....
.....
.....
.....
.....
.....
.....
...
.....
.....
...
.....
.....

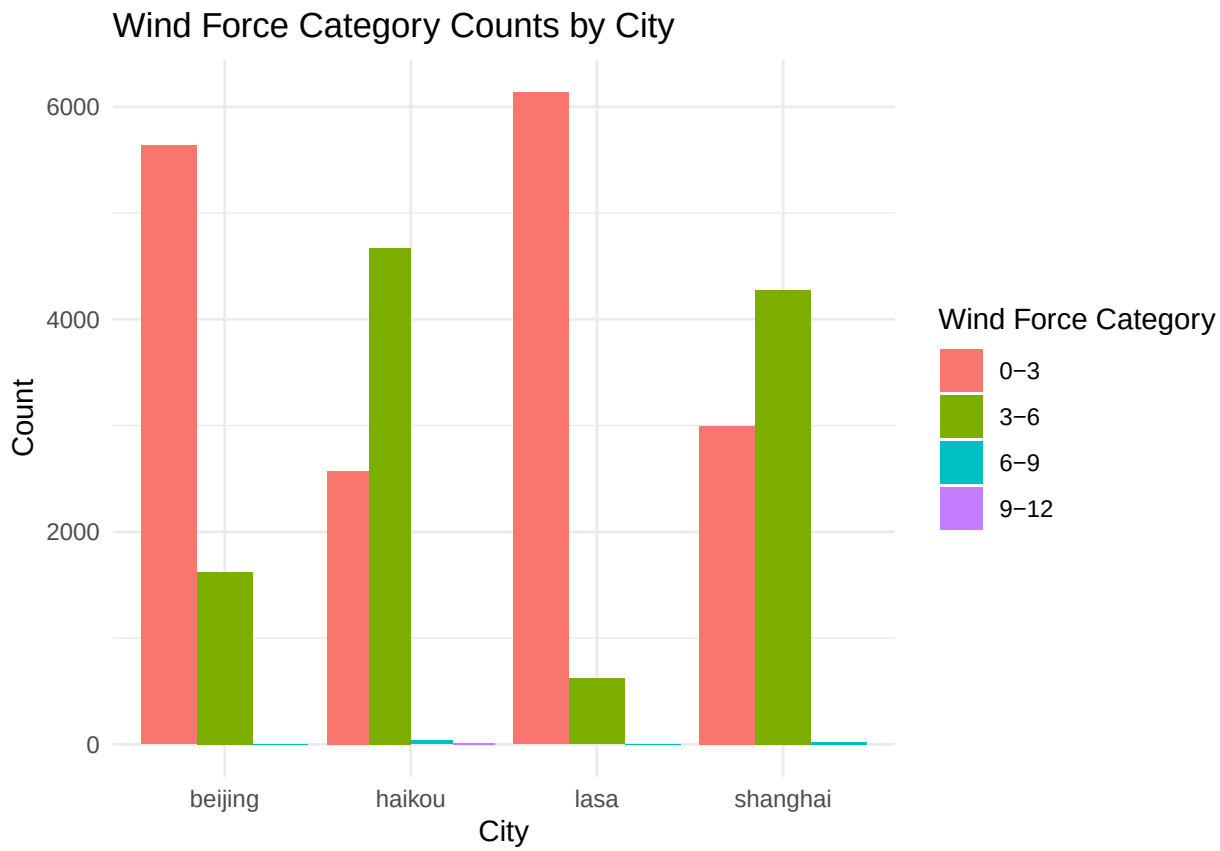
Night Weather Condition for haikou − Summer




Day Weather Condition for haikou − Autumn
Day.Weather
.....
.....
.....
.....
.....
.....
.....
.....
...
.....
...
.....
.....

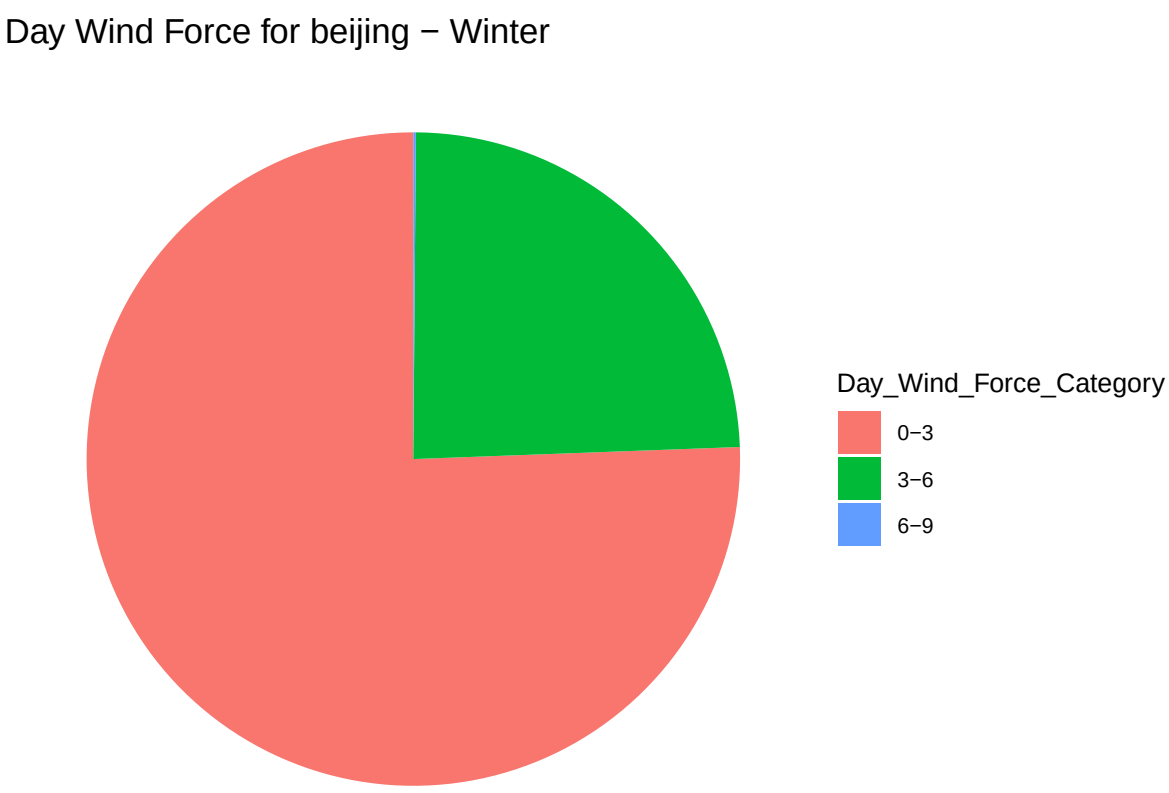


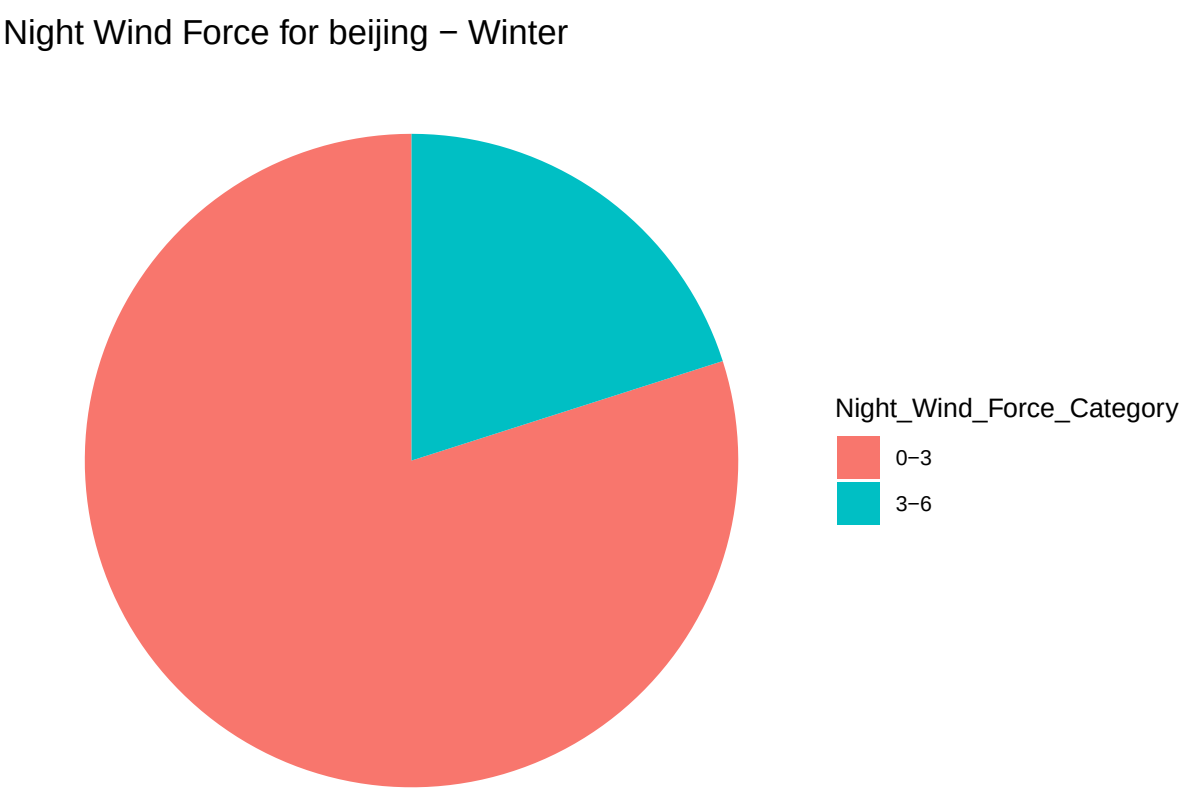

```
# Plot the value count of each city in the wind force categories
ggplot(df_long, aes(x = City, fill = Wind_Force_Category)) +
  geom_bar(position = "dodge") +
  labs(title = "Wind Force Category Counts by City",
       x = "City",
       y = "Count",
       fill = "Wind Force Category") +
  theme_minimal()
```

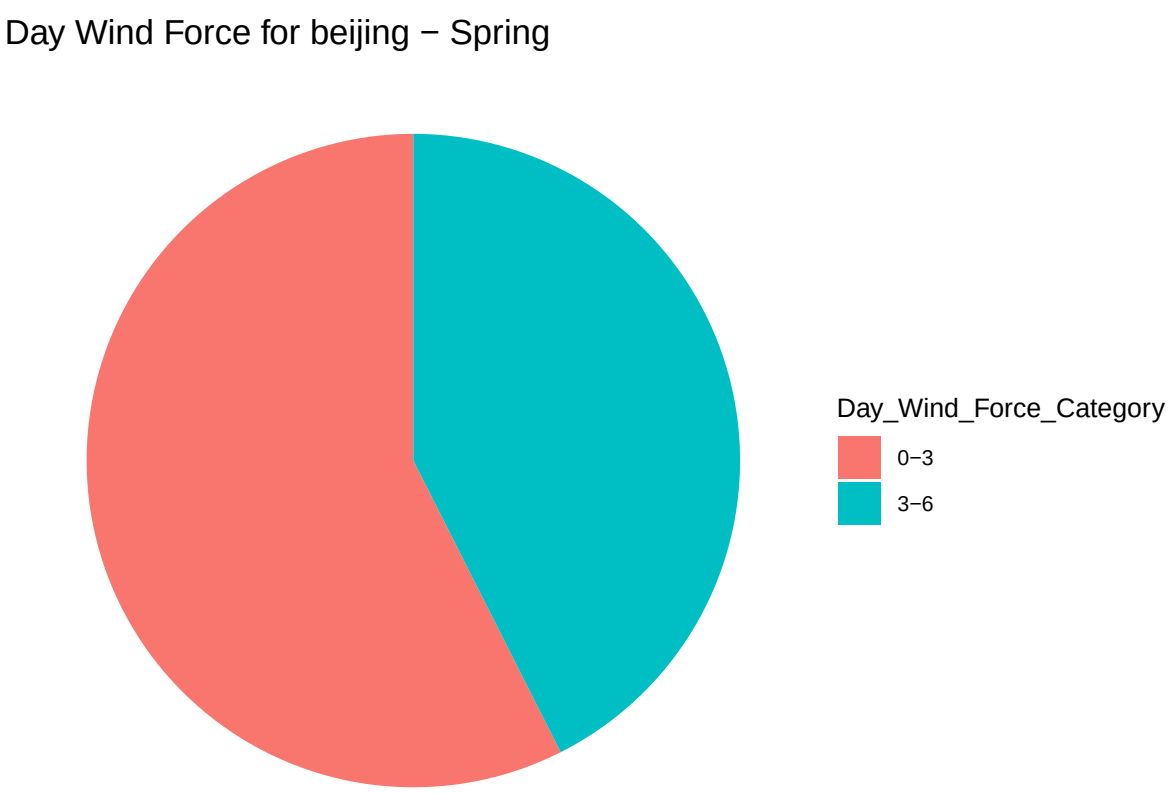


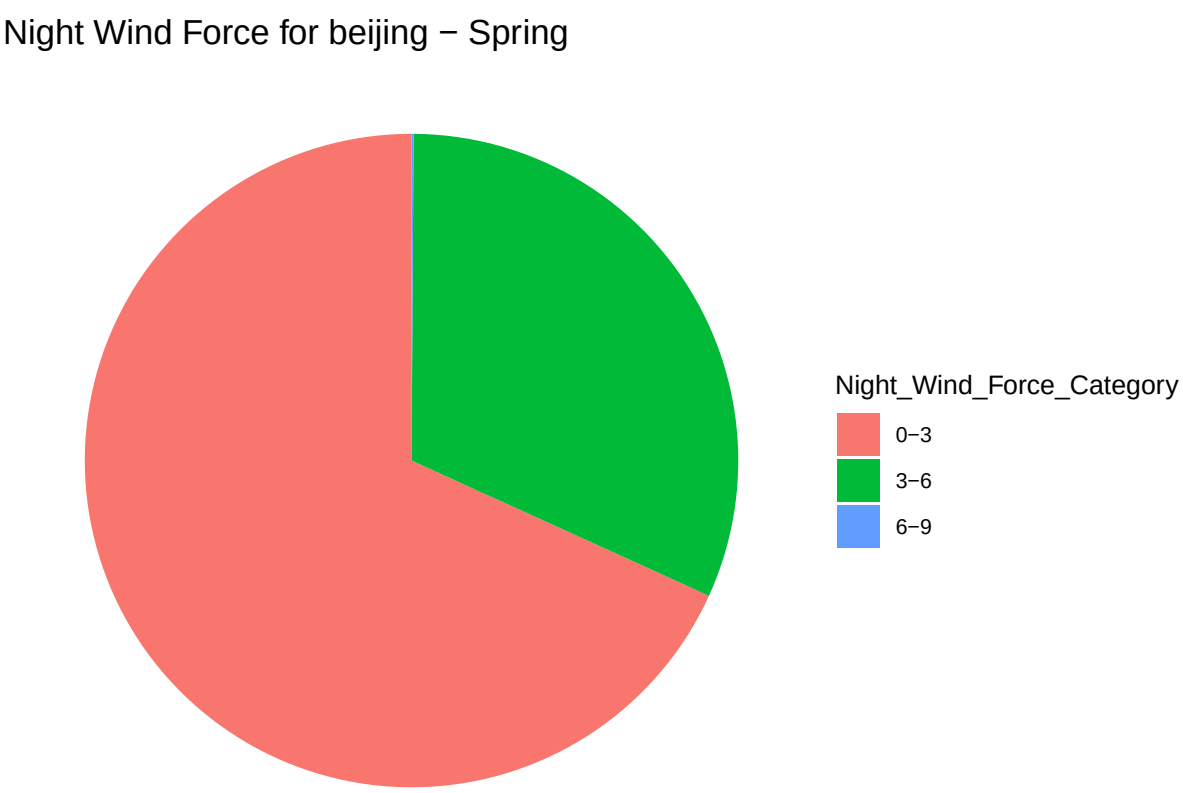

```r
# Count the occurrences of wind force by City and Season
day_wind_force_counts <- data %>%
  group_by(City, Season, Day_Wind_Force_Category) %>%
  tally() %>%
  ungroup()

night_wind_force_counts <- data %>%
  group_by(City, Season, Night_Wind_Force_Category) %>%
  tally() %>%
  ungroup()

# Function to plot pie chart for weather conditions or wind force by city and season
plot_pie_chart <- function(data, city, season, weather_column, title) {
  city_season_data <- data %>% filter(City == city & Season == season)
  p <- ggplot(city_season_data, aes(x = "", y = n, fill = !!sym(weather_column))) +
    geom_bar(stat = "identity", width = 1) +
    coord_polar(theta = "y") +
    labs(title = paste(title, "for", city, "-", season), fill = weather_column) +
    theme_void() +
    theme(legend.position = "right",  # Move legend outside the plot
          legend.title = element_text(size = 10), # Adjust the legend title size
          legend.text = element_text(size = 8))   # Adjust the legend text size
  return(p)
}

# Loop through cities and seasons to plot pie charts for Day and Night weather conditions and wind forc
```

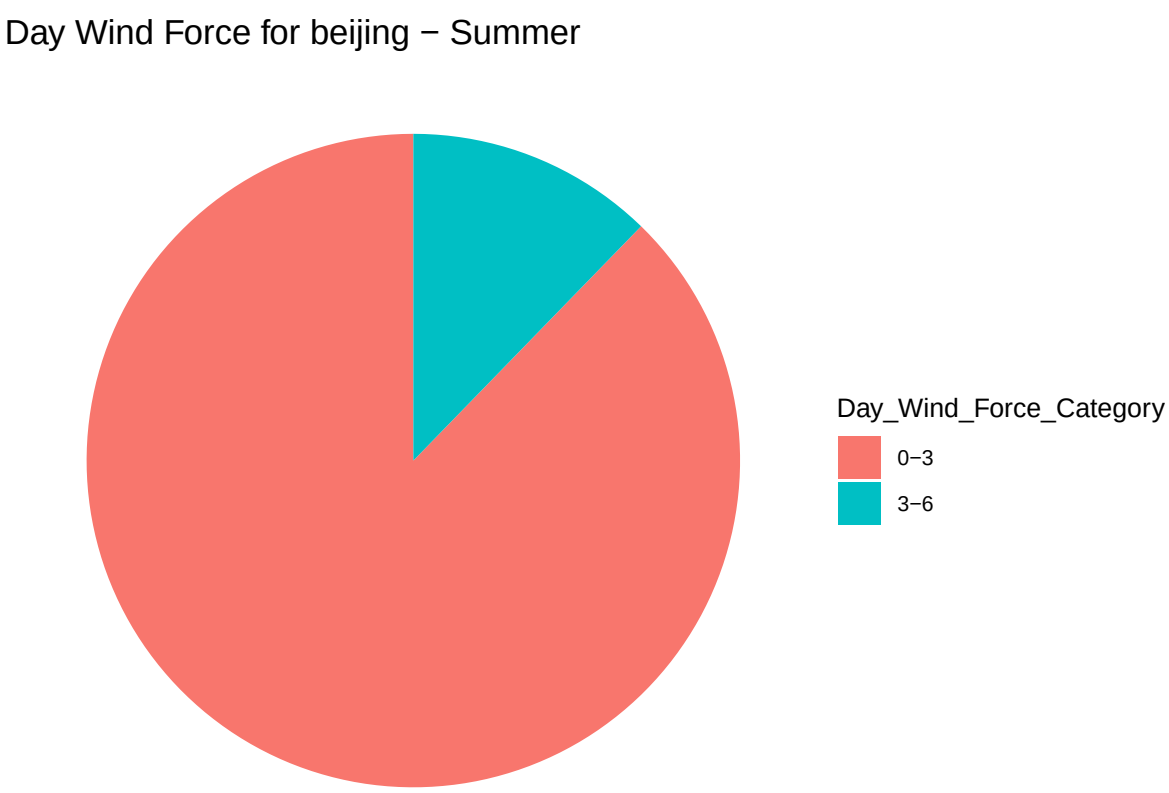
```r
cities <- unique(data$City)
seasons <- unique(data$Season)

for (city in cities) {
  for (season in seasons) {
    # Day Wind Force Pie Chart
    day_wind_plot <- plot_pie_chart(day_wind_force_counts, city, season, "Day_Wind_Force_Category", "Da
    print(day_wind_plot)

    # Night Wind Force Pie Chart
    night_wind_plot <- plot_pie_chart(night_wind_force_counts, city, season, "Night_Wind_Force_Category
    print(night_wind_plot)
  }
}
```

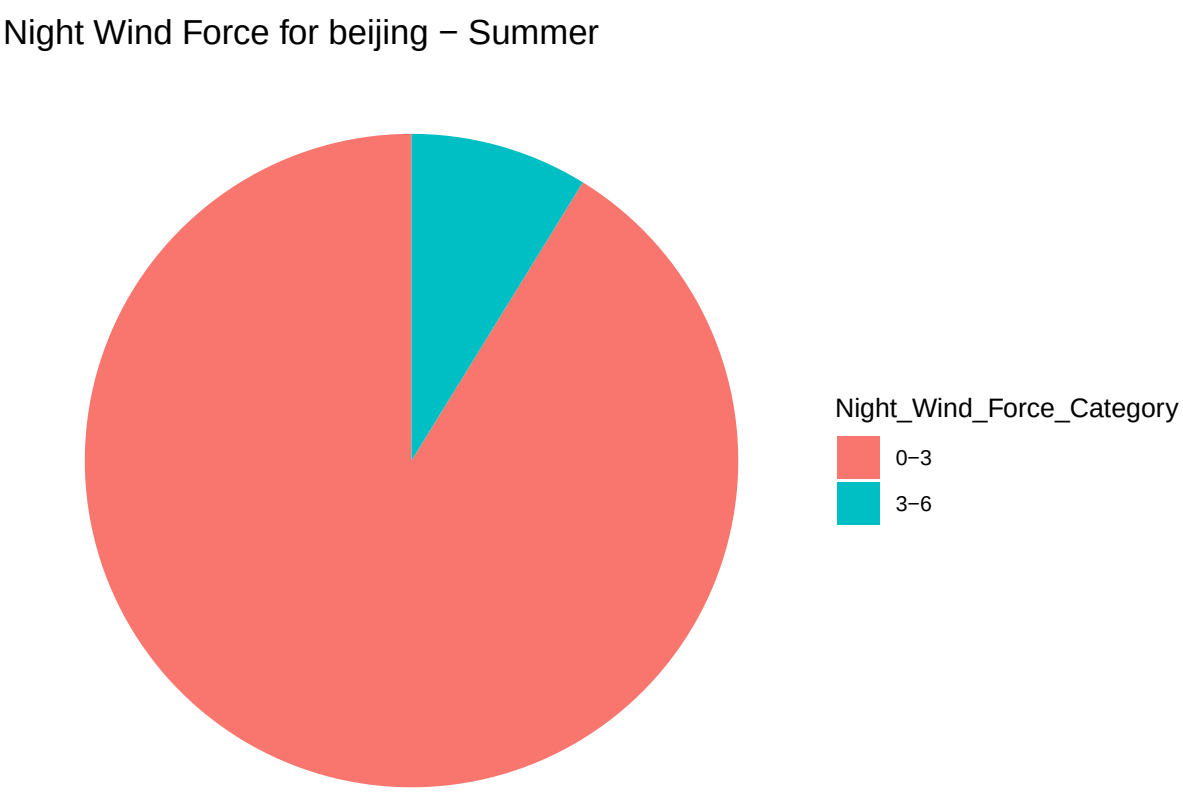


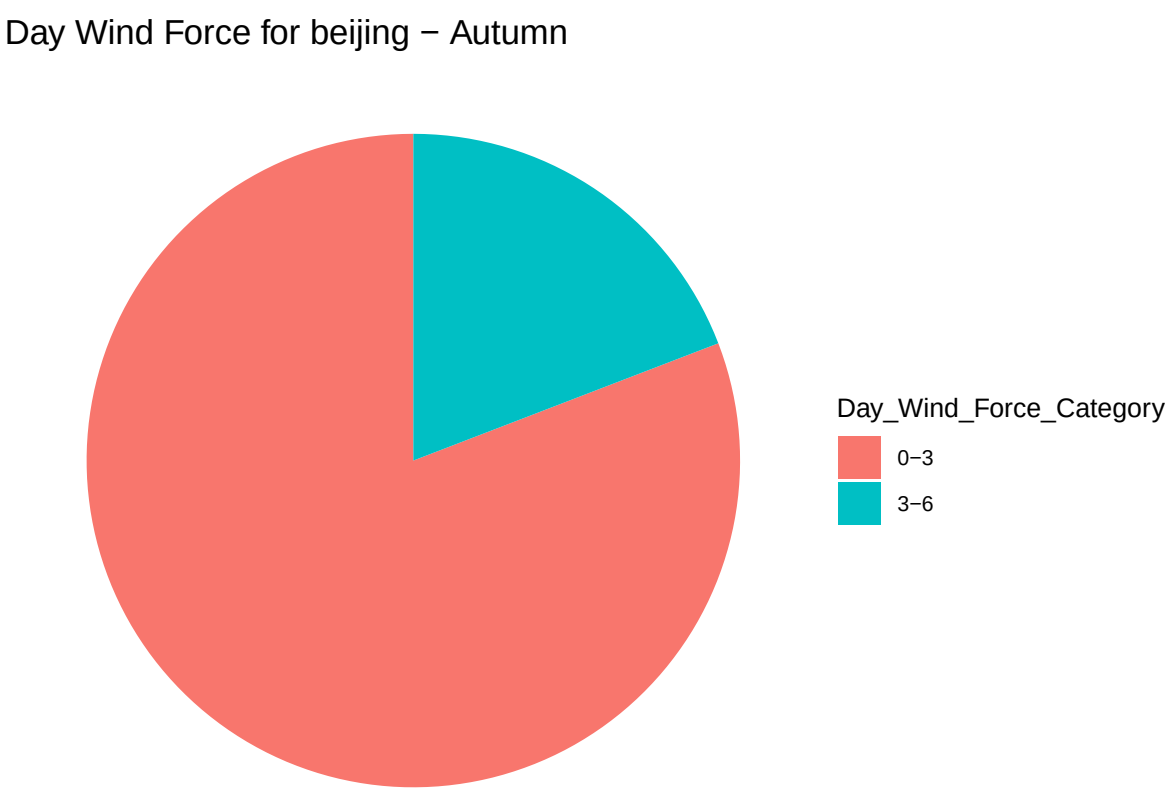

Night Wind Force for beijing − Winter
Night_Wind_Force_Category
0−3
3−6


Day Wind Force for beijing − Spring
Day_Wind_Force_Category
0−3
3−6

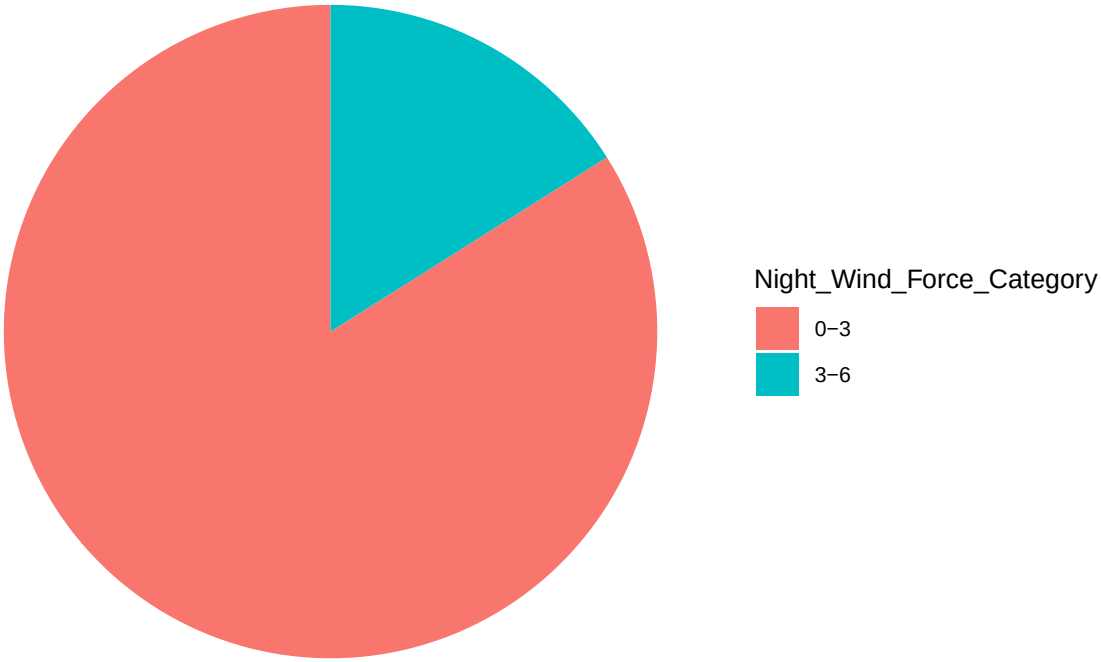

Night Wind Force for beijing − Spring
Night_Wind_Force_Category
0−3
3−6
6−9

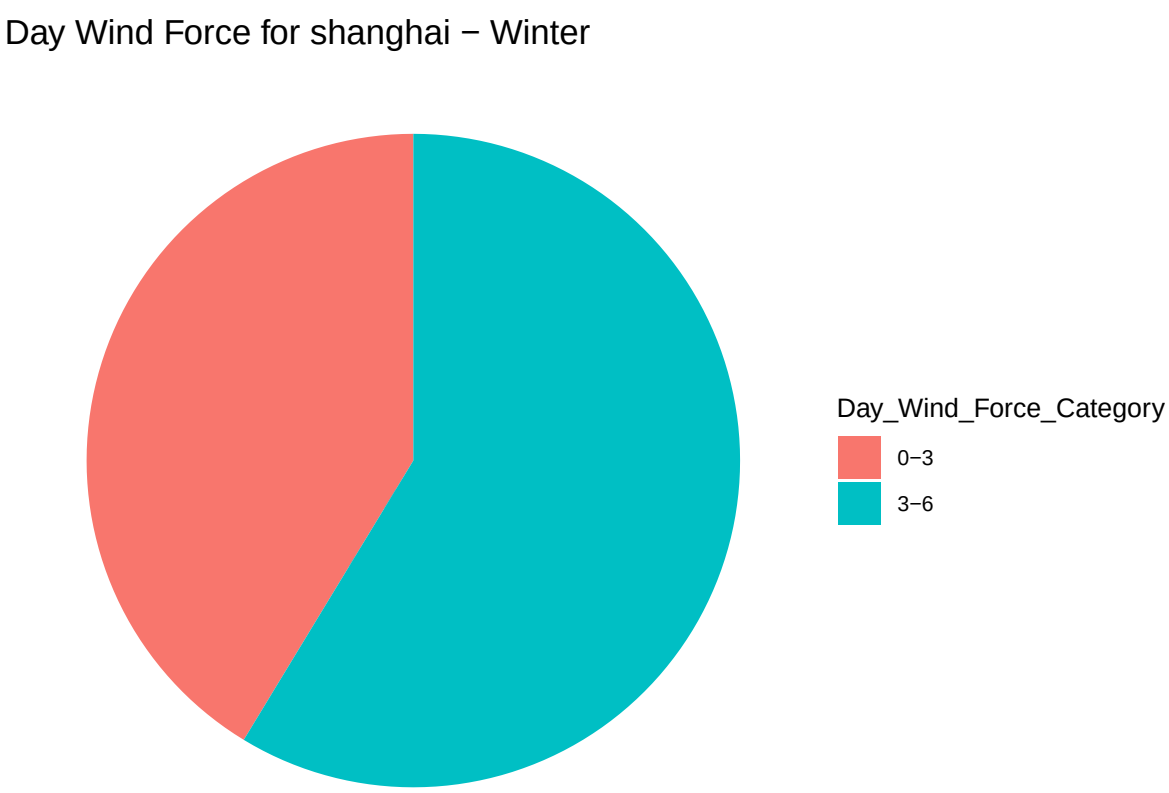

Day Wind Force for beijing − Summer
Day_Wind_Force_Category
0−3
3−6


Night Wind Force for beijing − Summer
Night_Wind_Force_Category
0−3
3−6

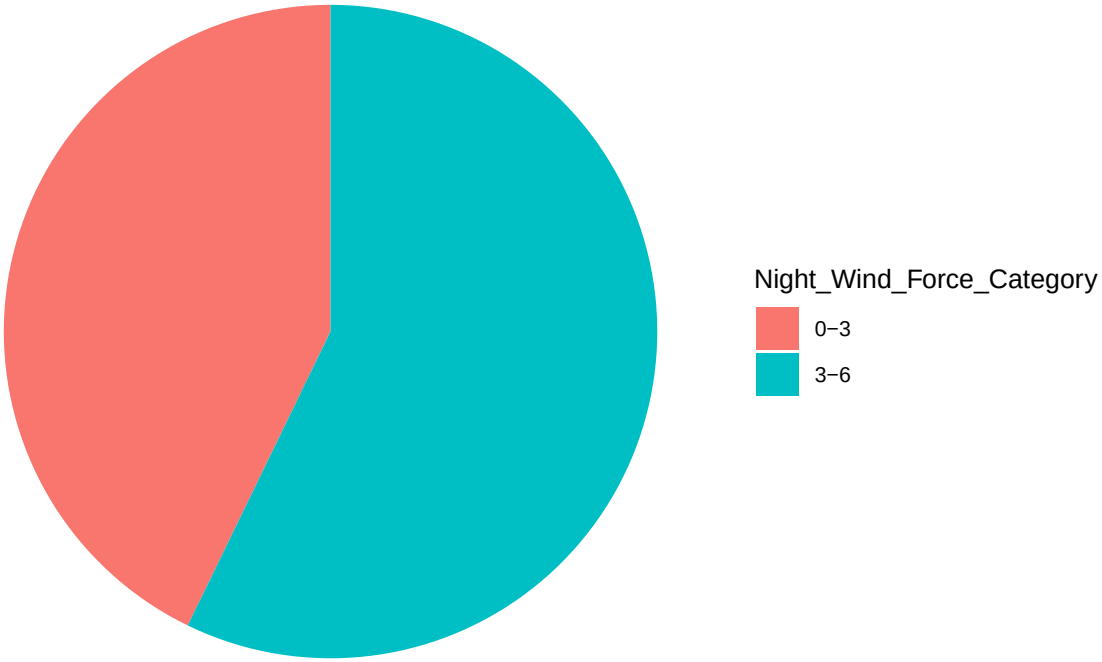

Day Wind Force for beijing − Autumn
Day_Wind_Force_Category
0−3
3−6

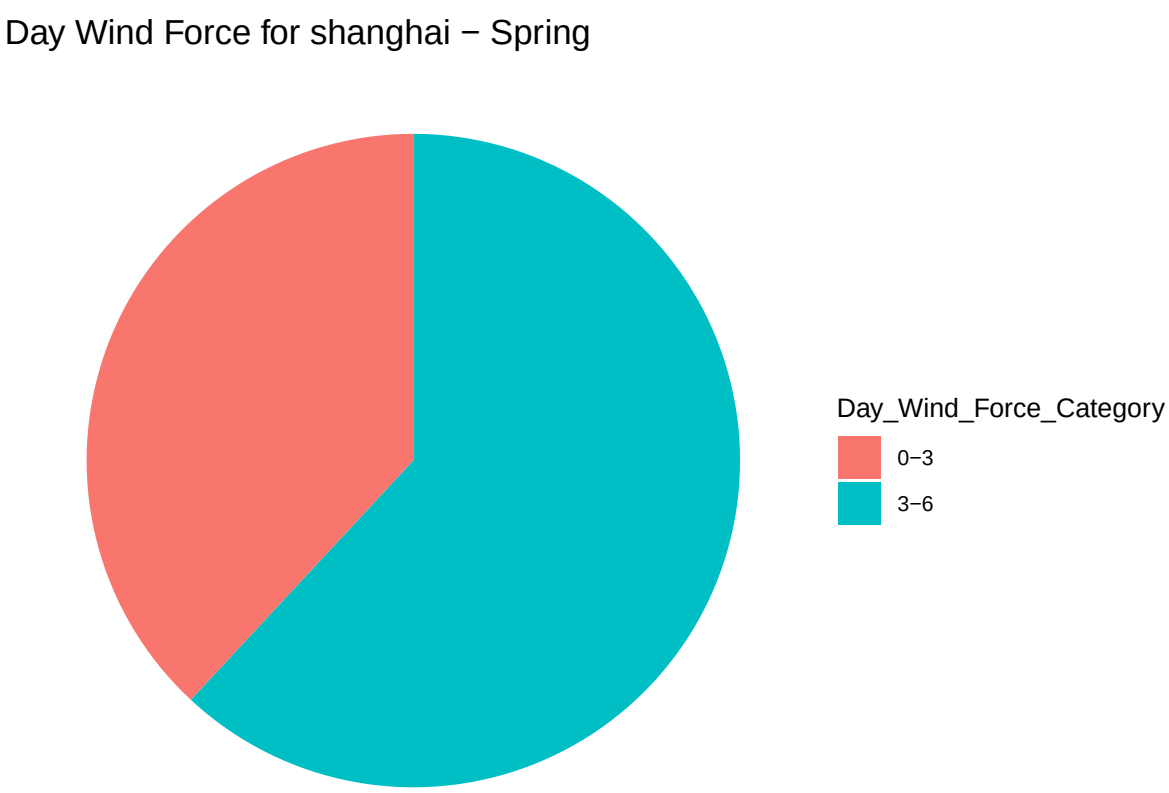
Night Wind Force for beijing − Autumn

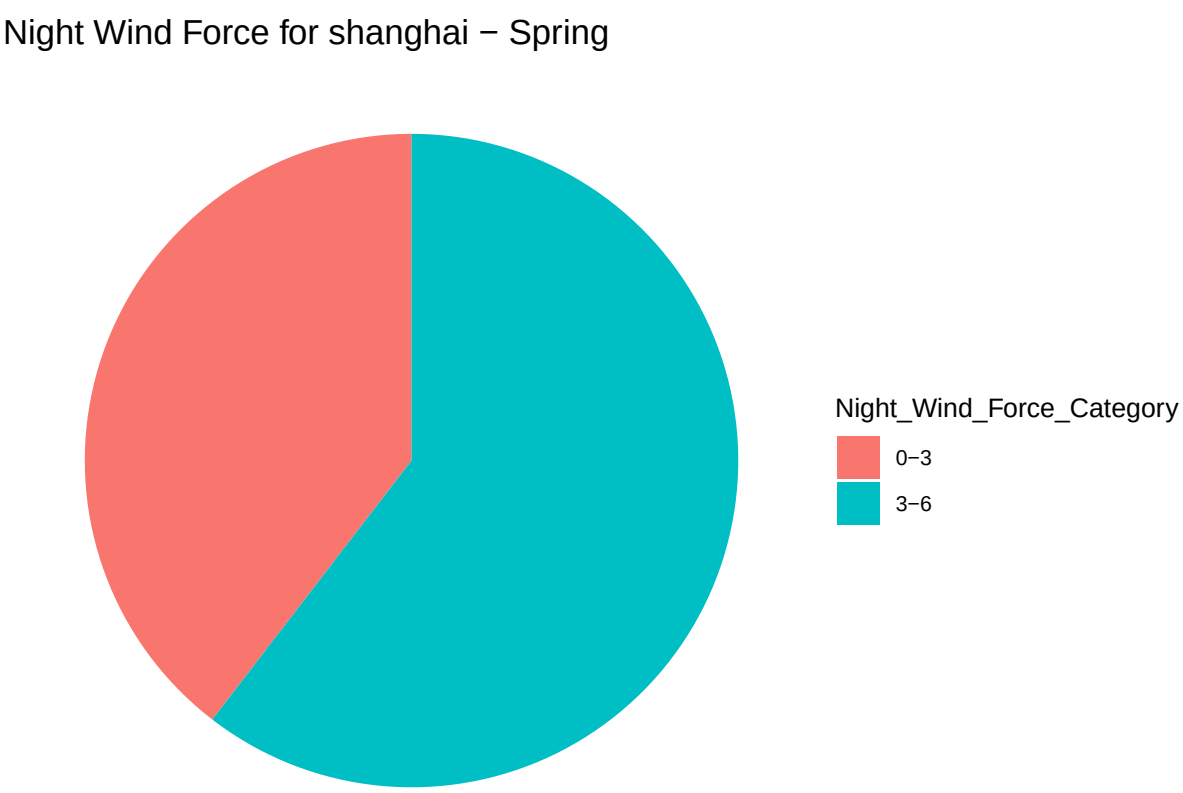


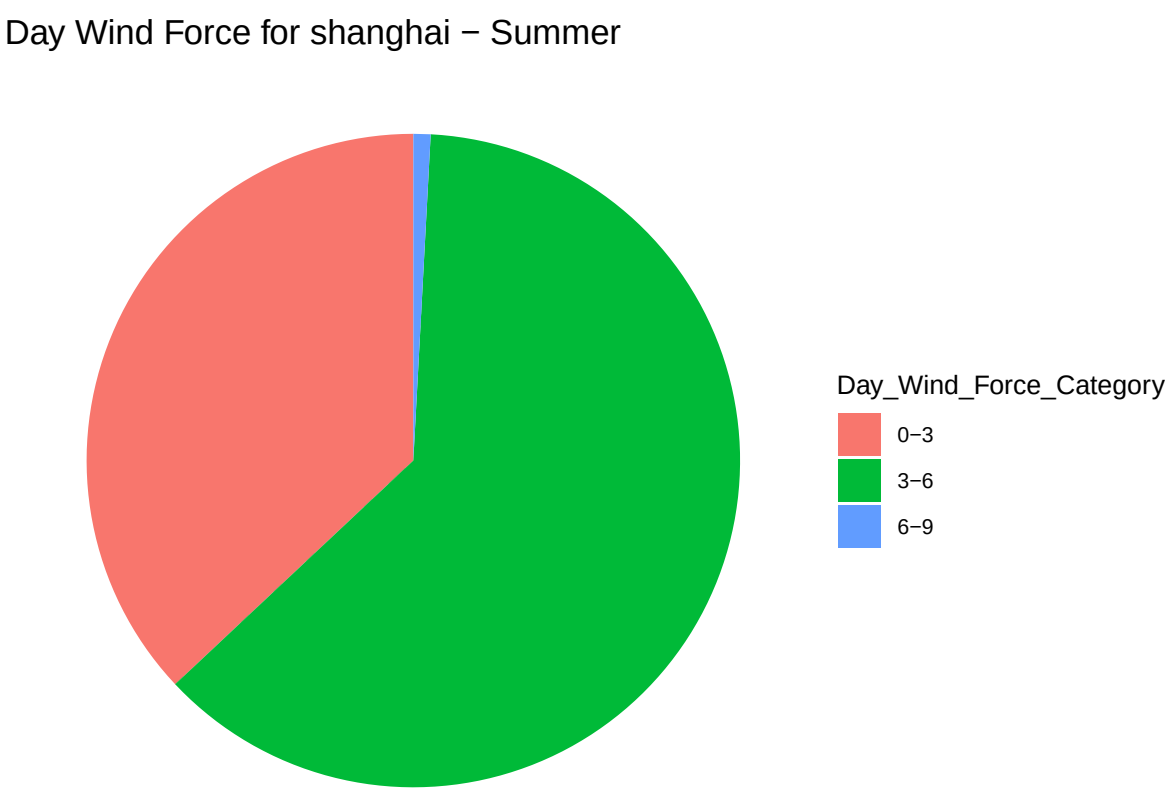

Day Wind Force for shanghai − Winter
Day_Wind_Force_Category
0−3
3−6

Night Wind Force for shanghai − Winter

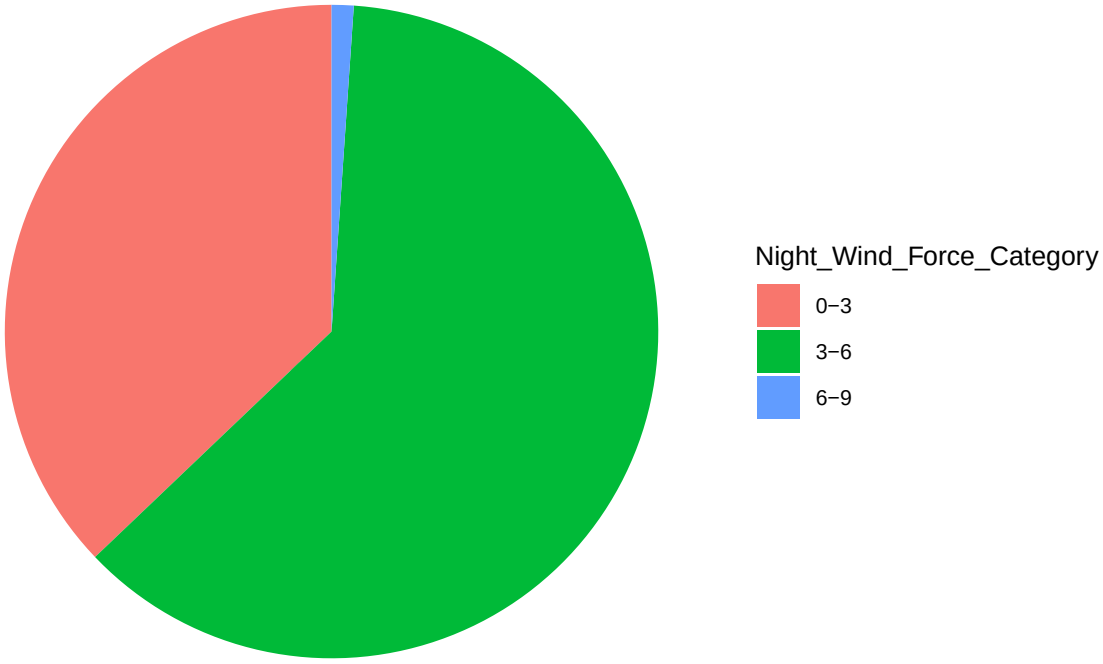


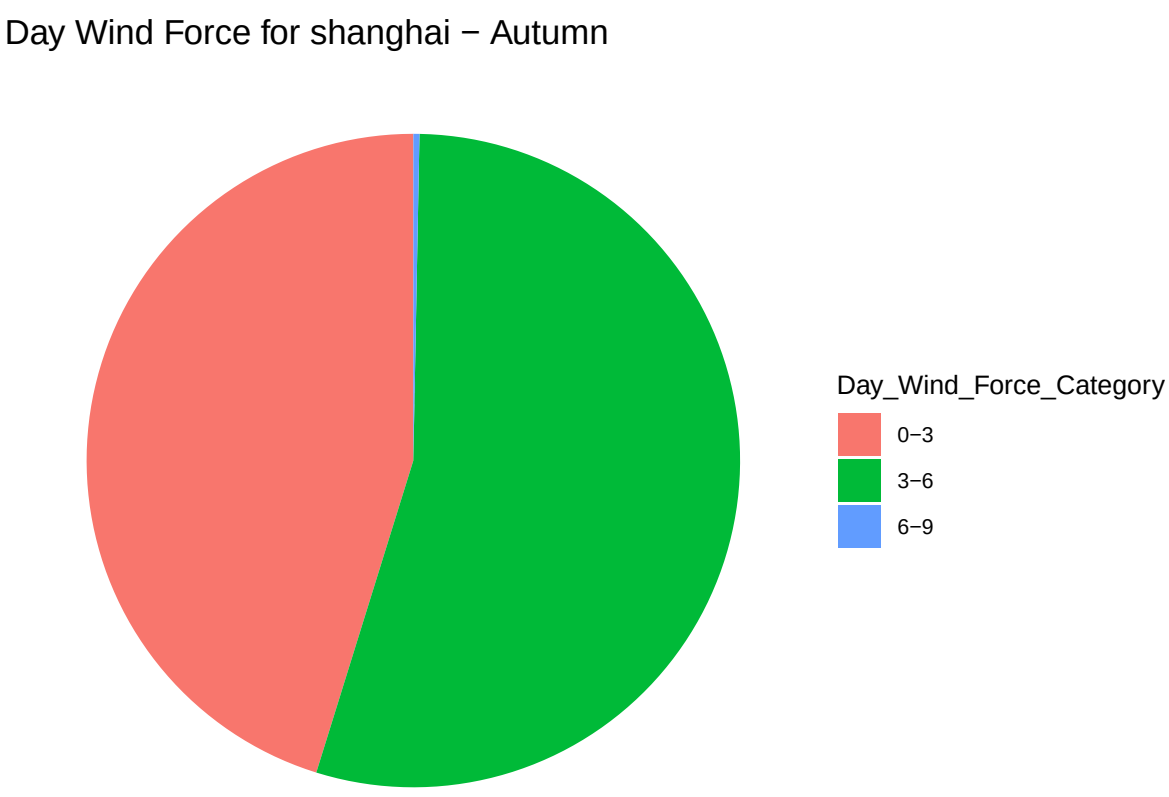

Day Wind Force for shanghai − Spring
Day_Wind_Force_Category
0−3
3−6

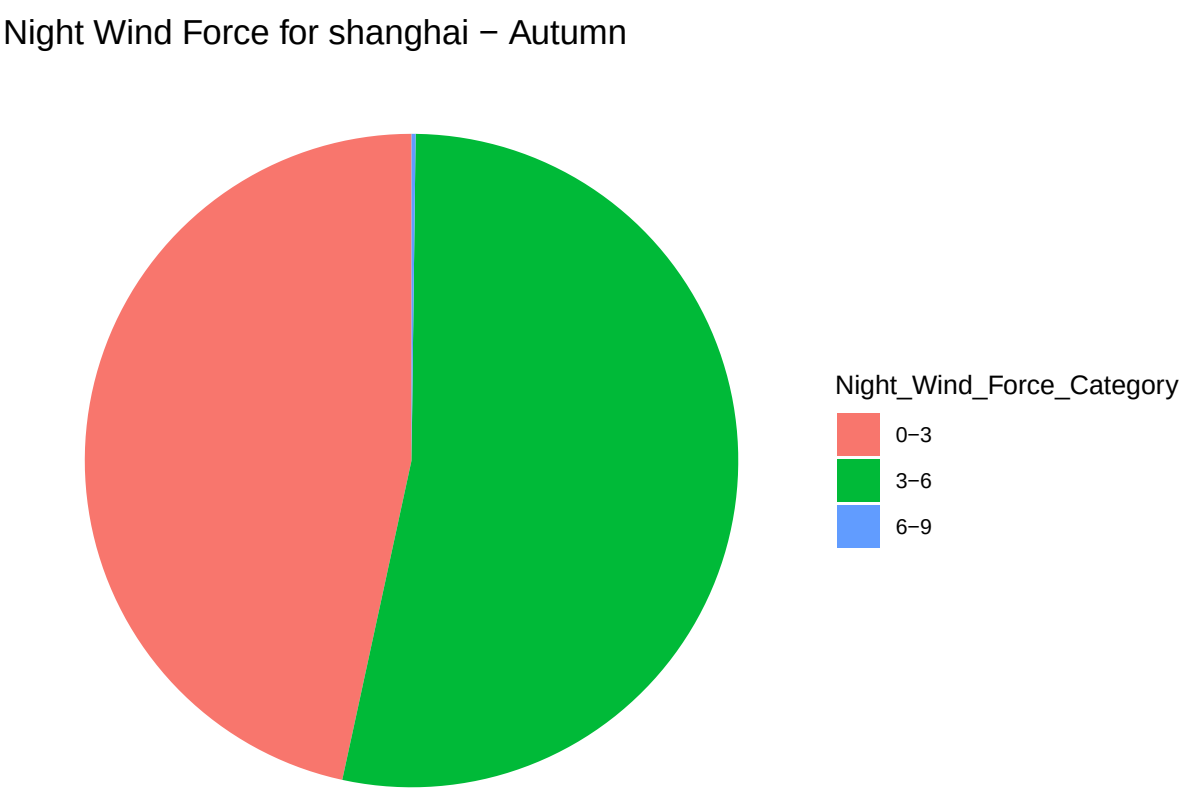

Night Wind Force for shanghai − Spring
Night_Wind_Force_Category
0−3
3−6


Day Wind Force for shanghai − Summer
Day_Wind_Force_Category
0−3
3−6
6−9

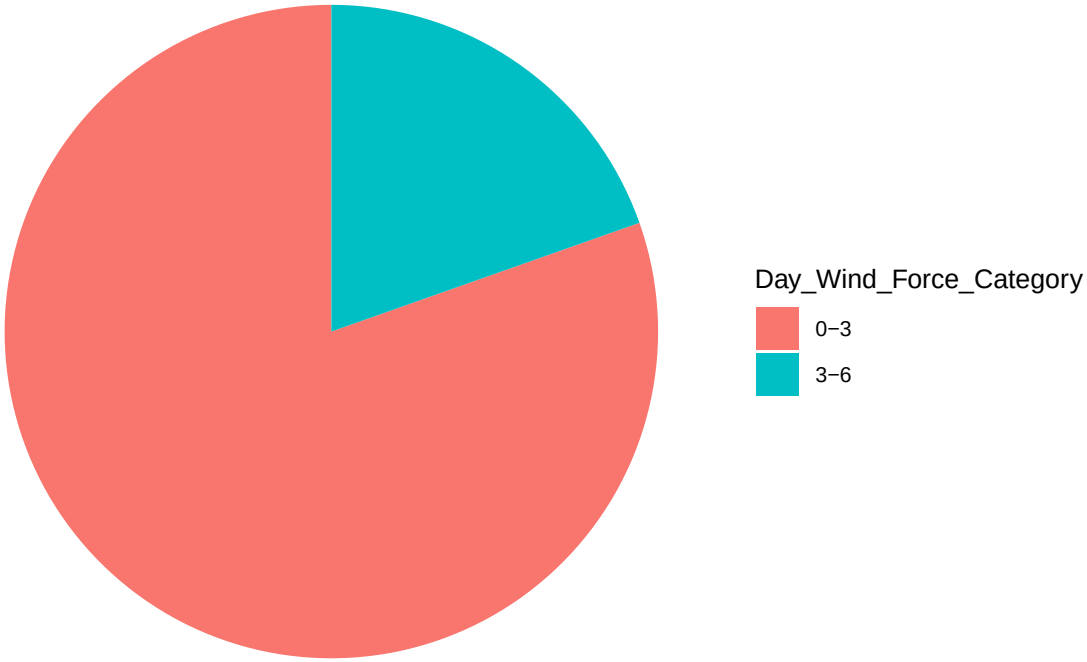
Night Wind Force for shanghai − Summer



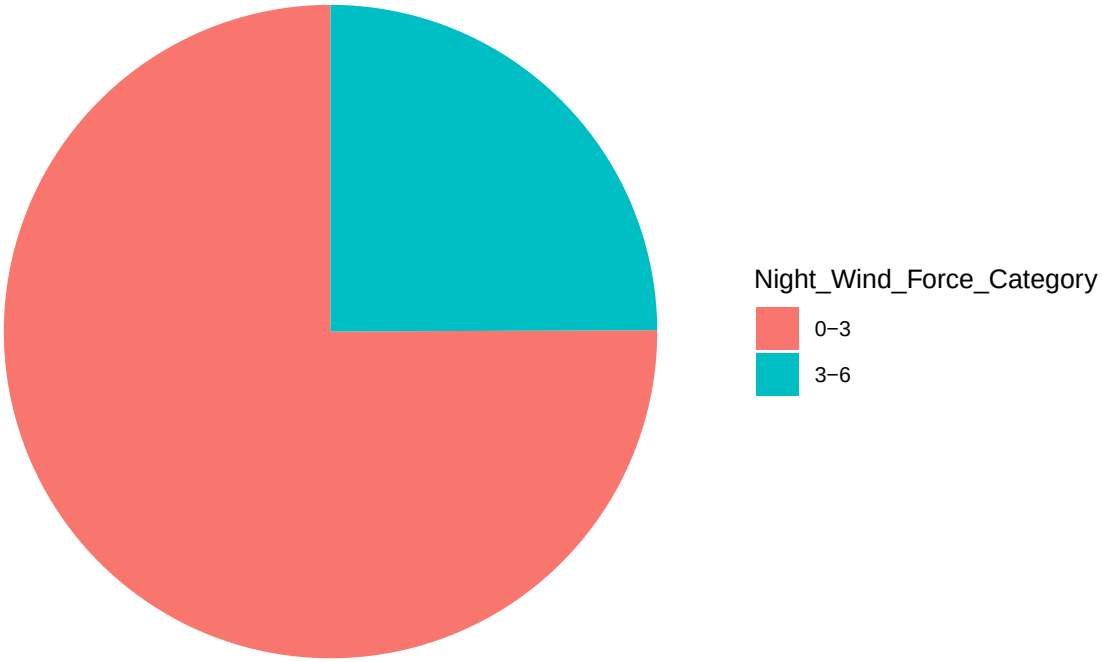

Day Wind Force for shanghai − Autumn
Day_Wind_Force_Category
0−3
3−6
6−9

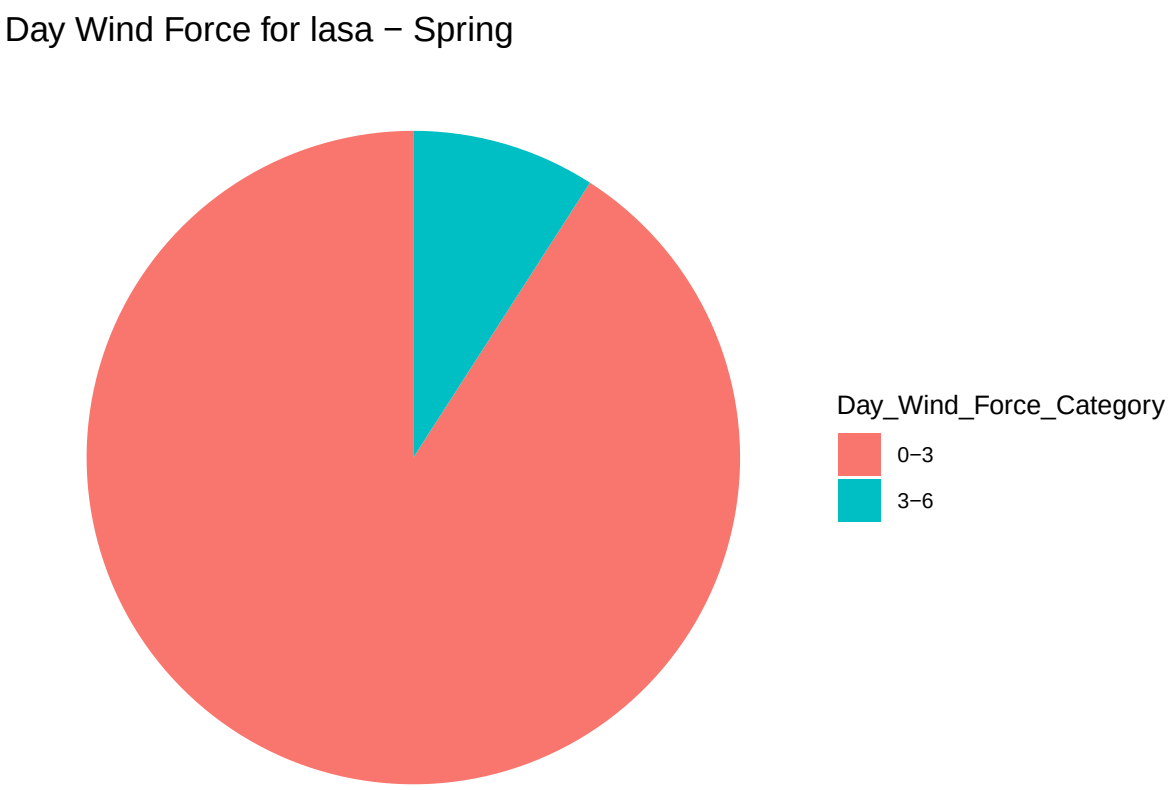

Night Wind Force for shanghai − Autumn
Night_Wind_Force_Category
0−3
3−6
6−9

Day Wind Force for lasa − Winter

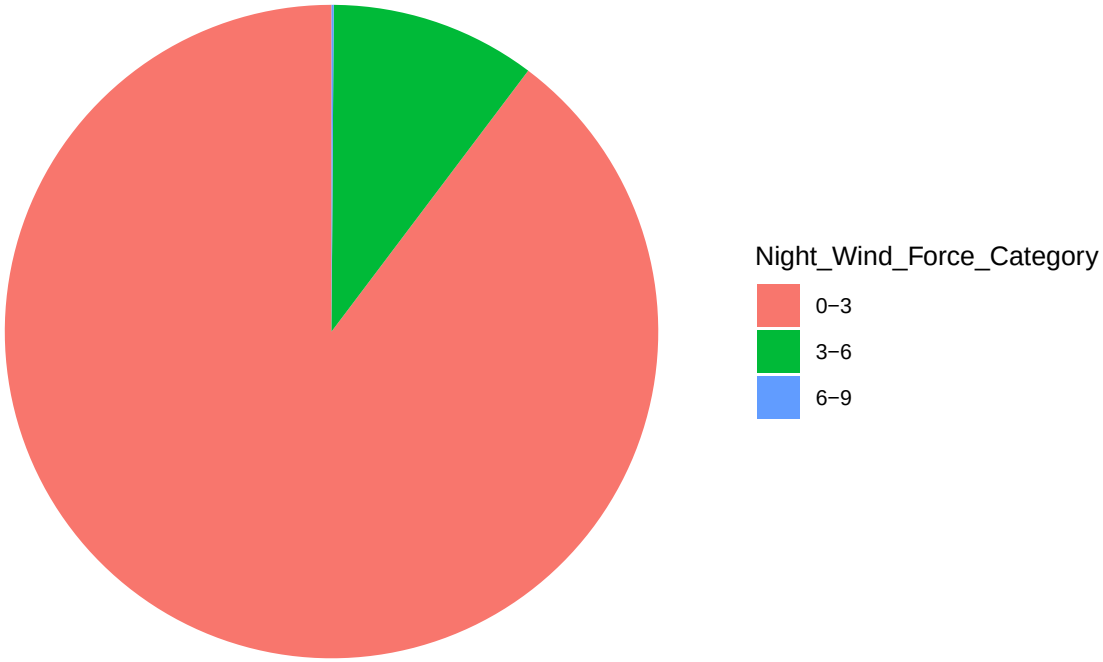


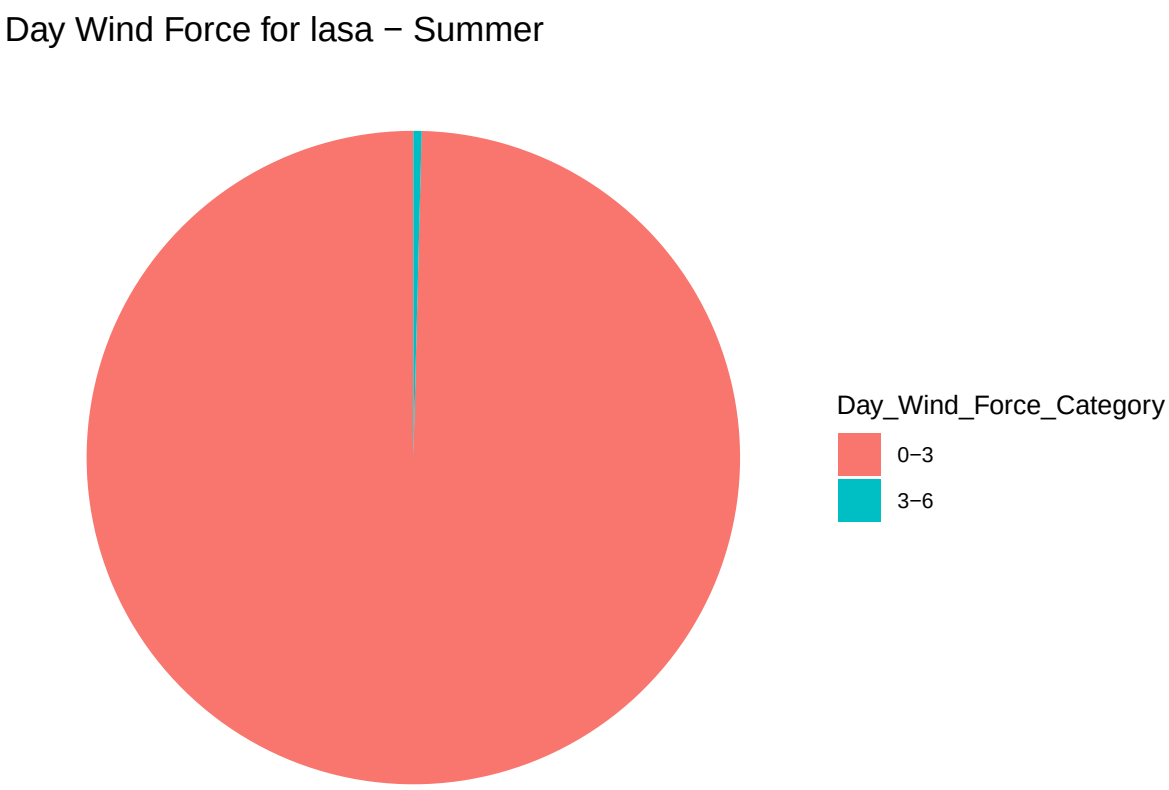
Night Wind Force for lasa − Winter

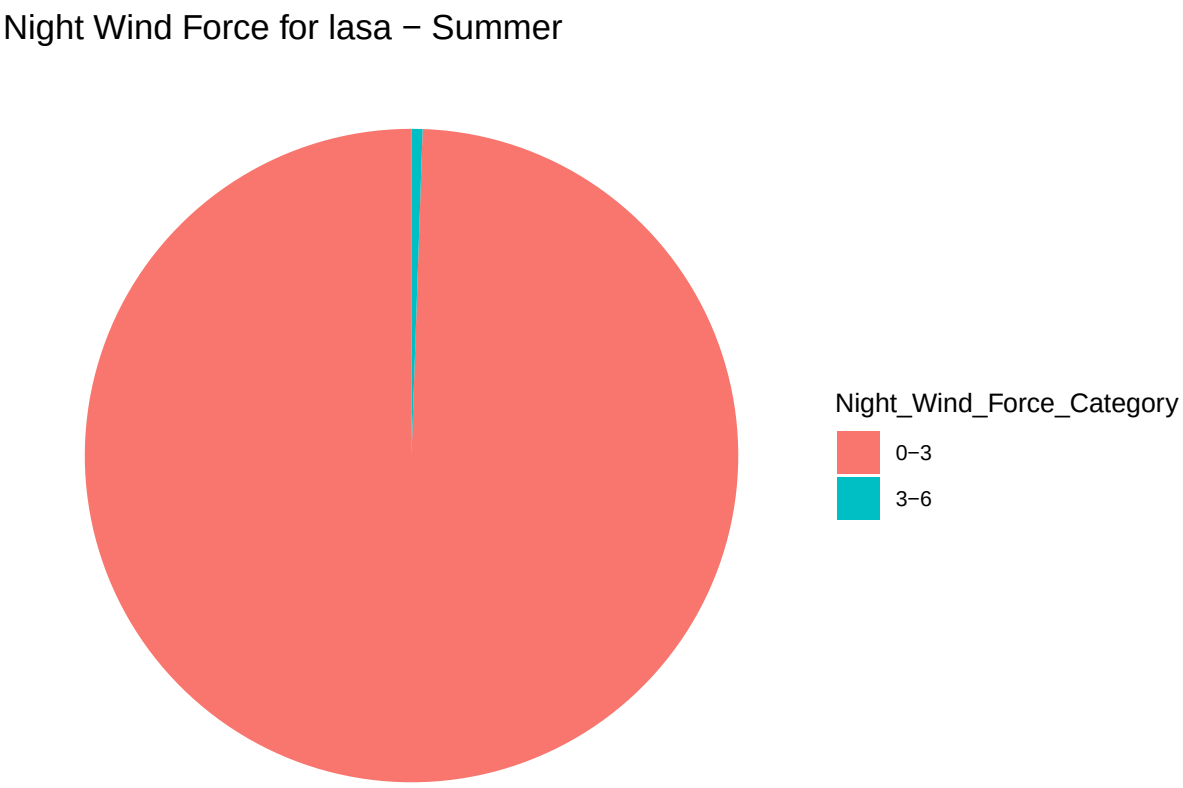


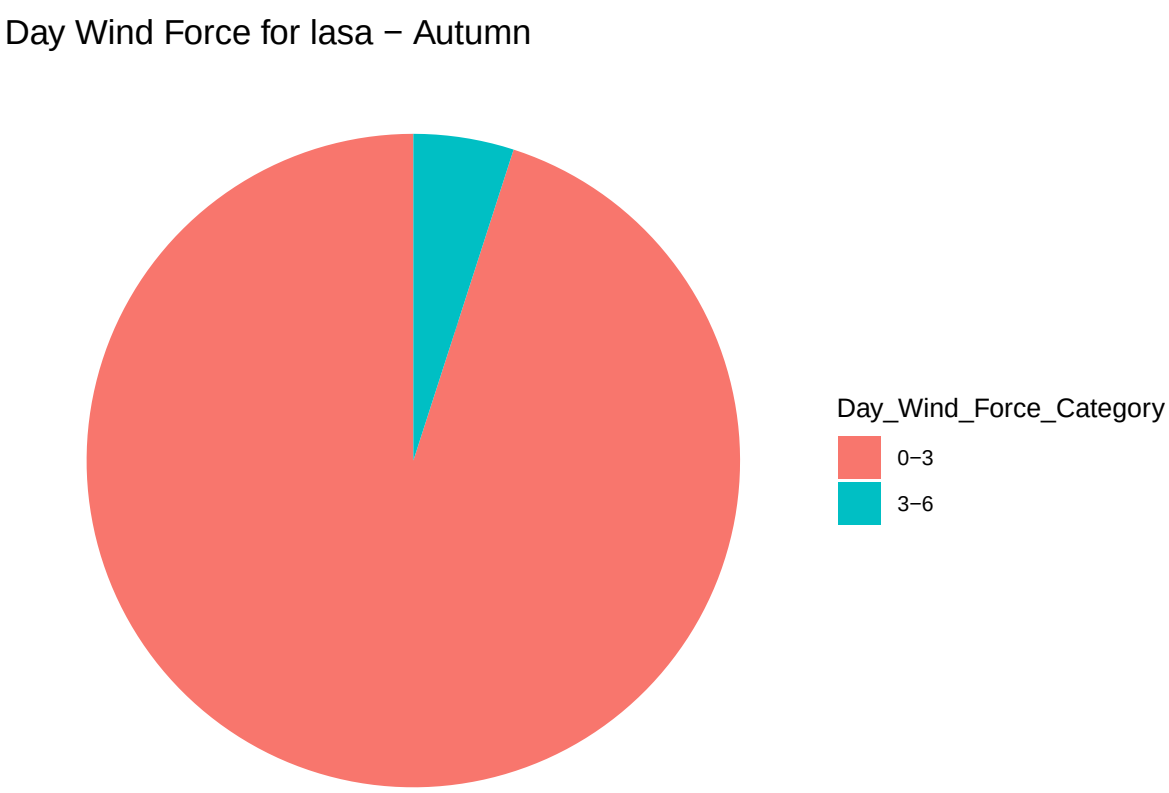

Day Wind Force for lasa − Spring
Day_Wind_Force_Category
0−3
3−6

Night Wind Force for lasa − Spring

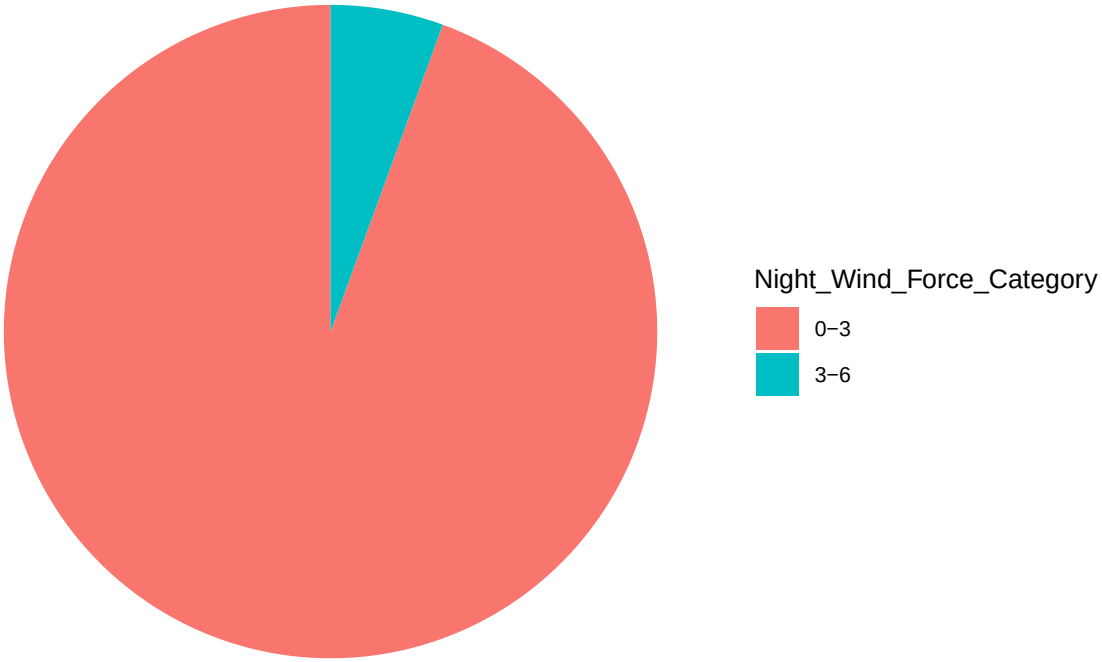



Day Wind Force for lasa − Summer
Day_Wind_Force_Category
0−3
3−6

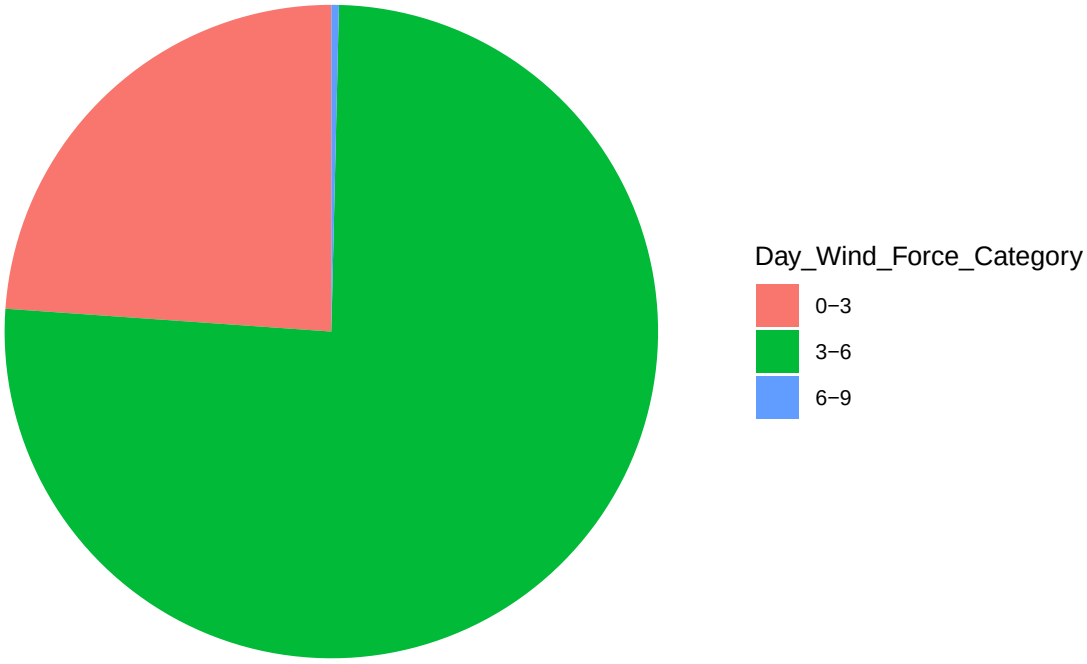

Night Wind Force for lasa − Summer
Night_Wind_Force_Category
0−3
3−6


Day Wind Force for lasa − Autumn
Day_Wind_Force_Category
0−3
3−6

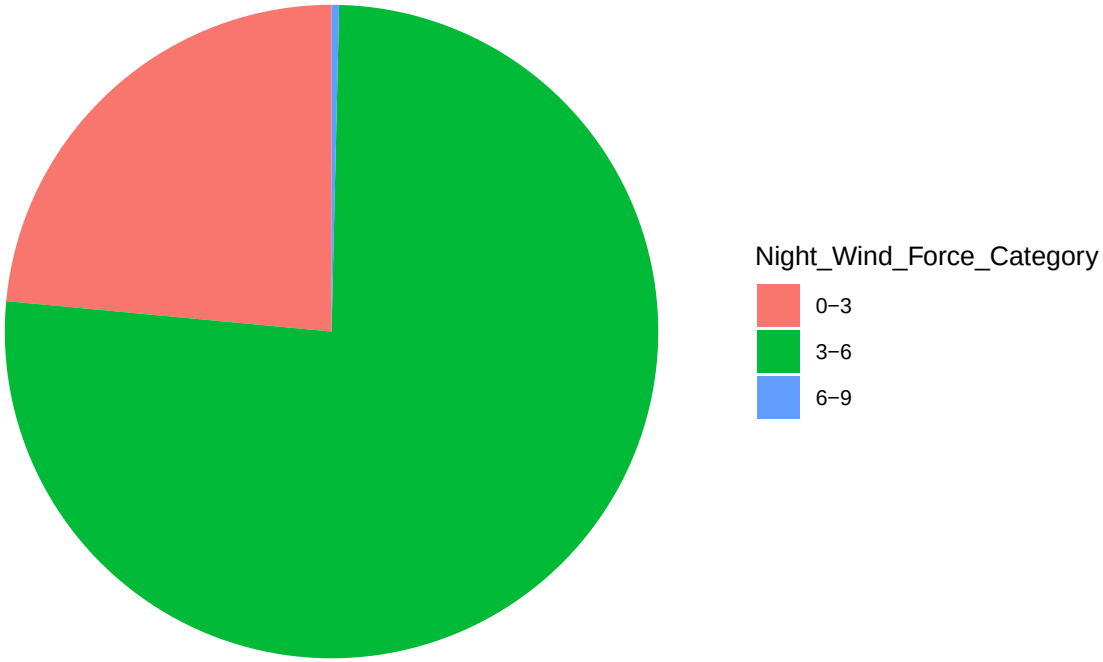
Night Wind Force for lasa − Autumn

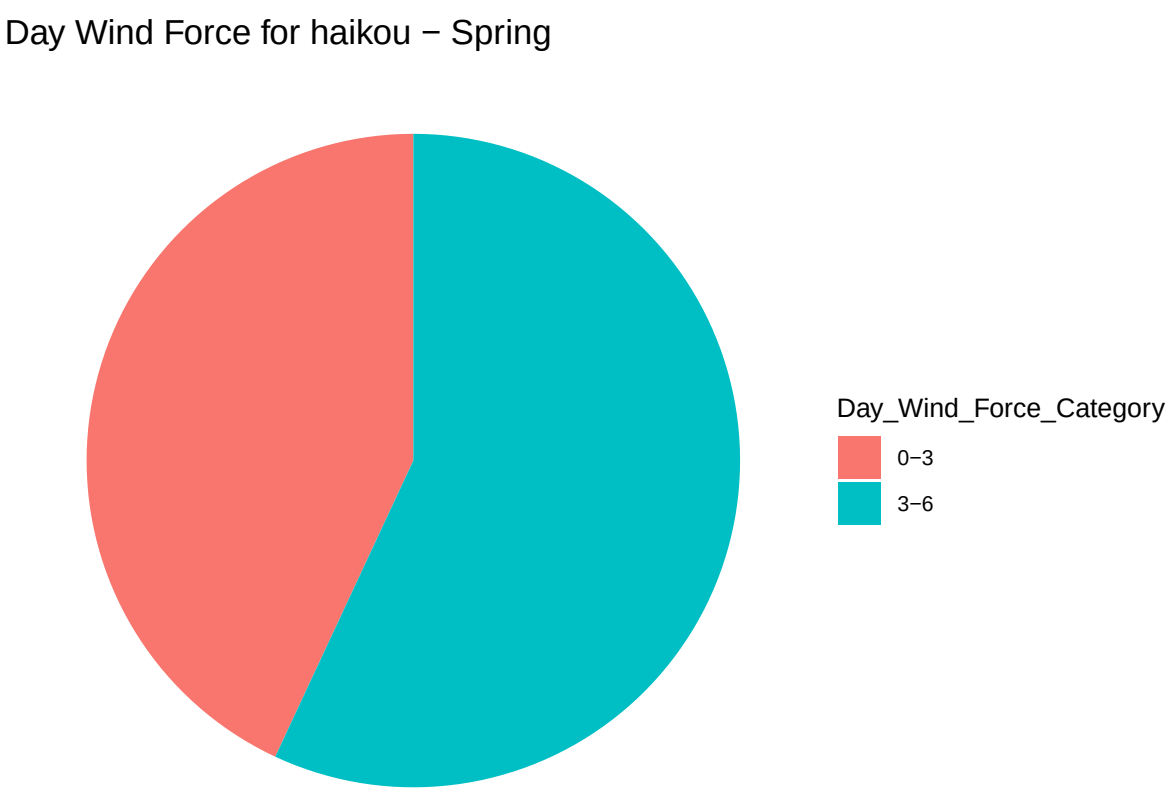


Day Wind Force for haikou − Winter

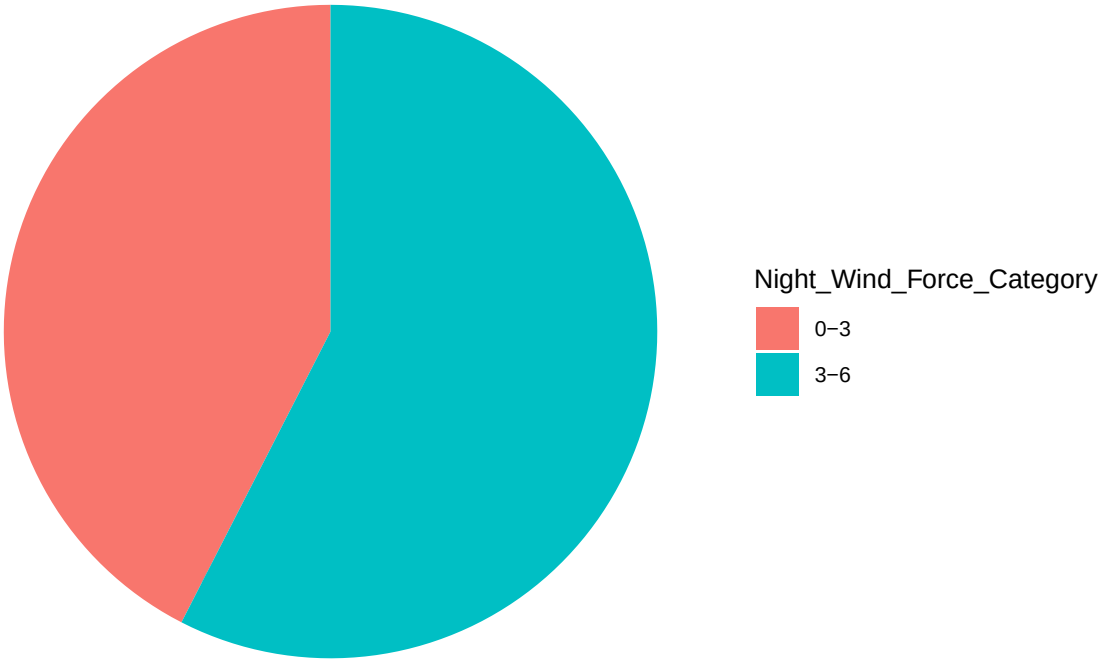


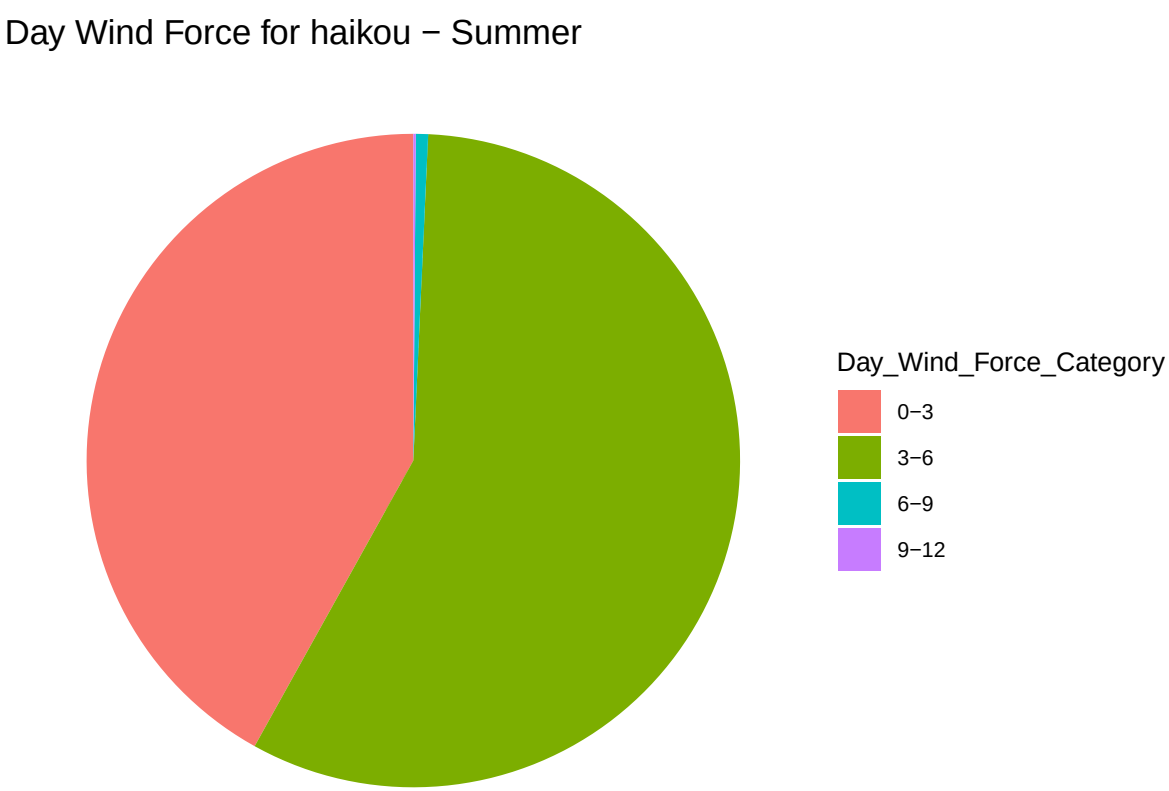
Night Wind Force for haikou − Winter

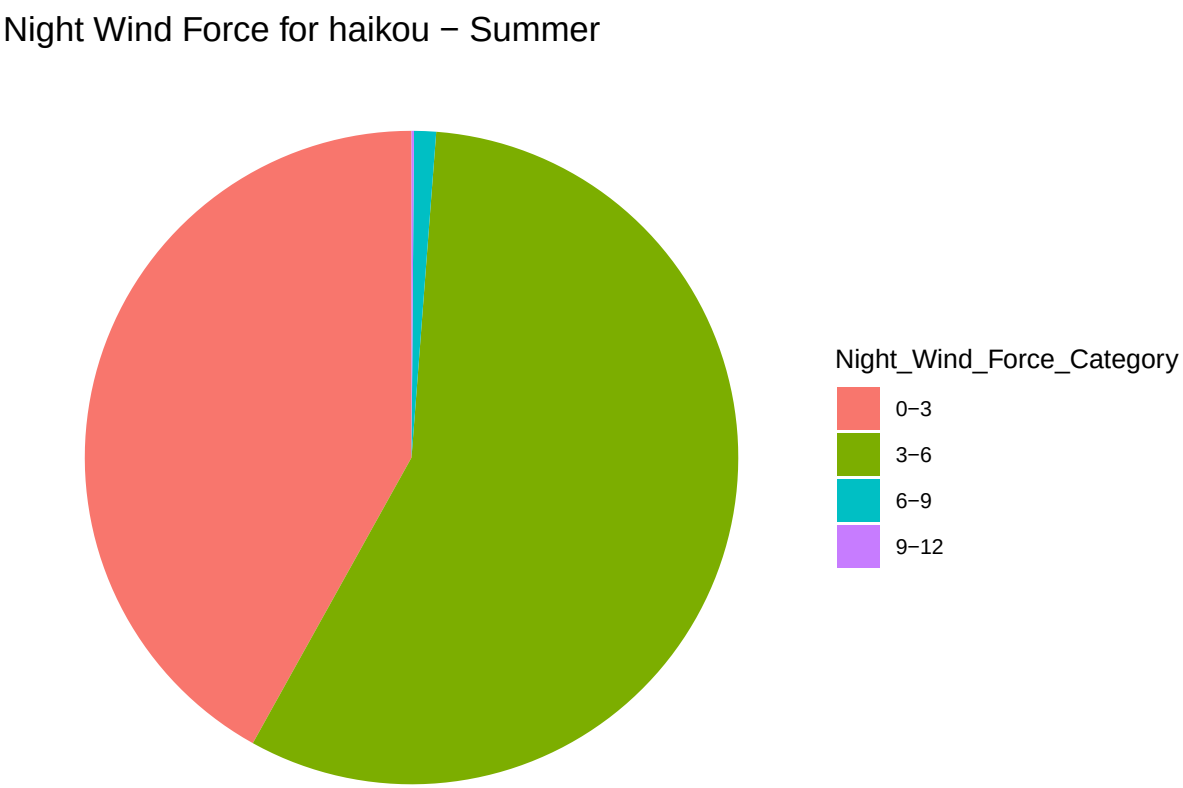


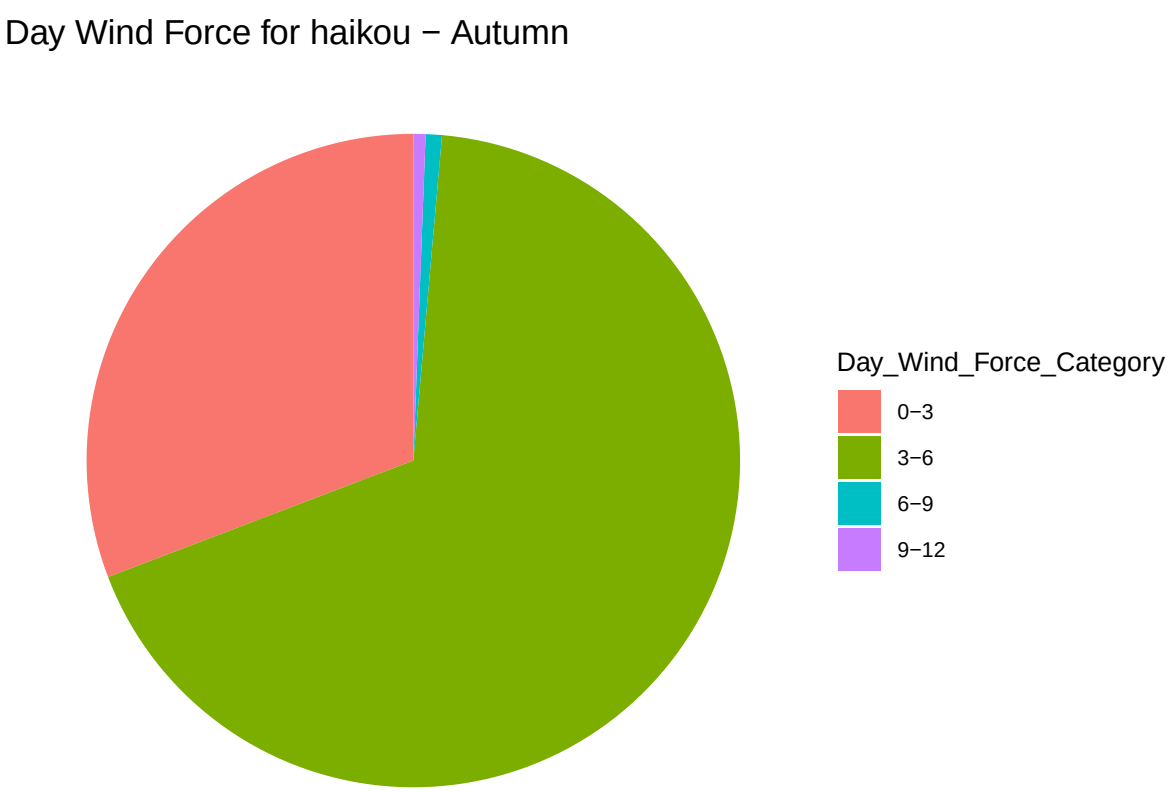

Day Wind Force for haikou − Spring
Day_Wind_Force_Category
0−3
3−6

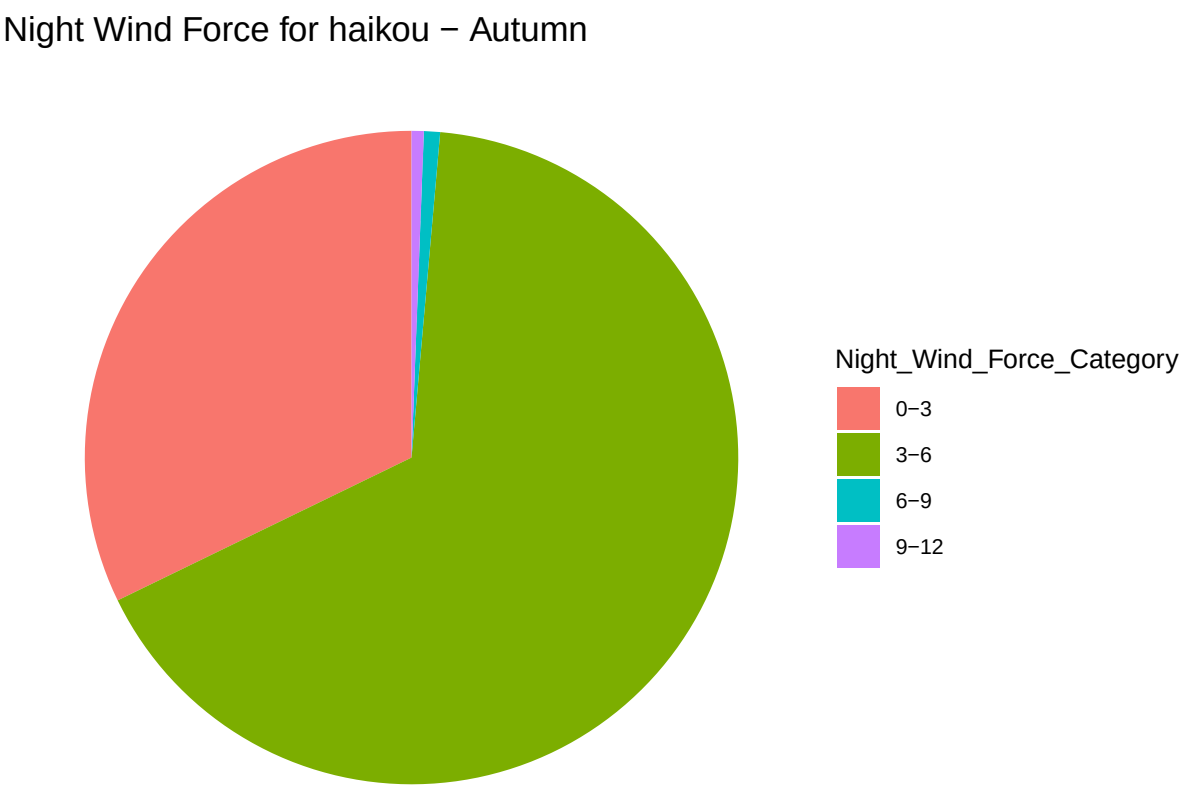
Night Wind Force for haikou − Spring




Day Wind Force for haikou − Summer
Day_Wind_Force_Category
0−3
3−6
6−9
9−12


Night Wind Force for haikou − Summer
Night_Wind_Force_Category
0−3
3−6
6−9
9−12


Day Wind Force for haikou − Autumn
Day_Wind_Force_Category
0−3
3−6
6−9
9−12


Night Wind Force for haikou − Autumn
Night_Wind_Force_Category
0−3
3−6
6−9
9−12