

### A. 3 gráficos generados con Plotly Express

```
# Instalando en el notebook via pip
!pip install plotly_express
```

🔄 Installing collected packages: plotly\_express  
Successfully installed plotly\_express-0.4.1

```
# Importando librerias
import plotly_express as px
```

```
# En este caso usaremos la que nos da el plotly, la cual trata sobre autos
data_car=px.data.carshare()
```

```
# Añadimos una categoría según la hora pico - madrugada (0-6), mañana (7-12), tarde (13-18), noche (19-23)
def categorizar_hora(h):
    if 0 <= h <= 6:
        return 'Madrugada'
    elif 7 <= h <= 12:
        return 'Mañana'
    elif 13 <= h <= 18:
        return 'Tarde'
    else:
        return 'Noche'
```

```
data_car['franja_horaria'] = data_car['peak_hour'].apply(categorizar_hora)
```

```
# Comprobamos el contenido del dataset
data_car.head()
```

🔄

	centroid_lat	centroid_lon	car_hours	peak_hour	franja_horaria
0	45.471549	-73.588684	1772.750000	2	Madrugada
1	45.543865	-73.562456	986.333333	23	Noche
2	45.487640	-73.642767	354.750000	20	Noche
3	45.522870	-73.595677	560.166667	23	Noche
4	45.453971	-73.738946	2836.666667	19	Noche

📊

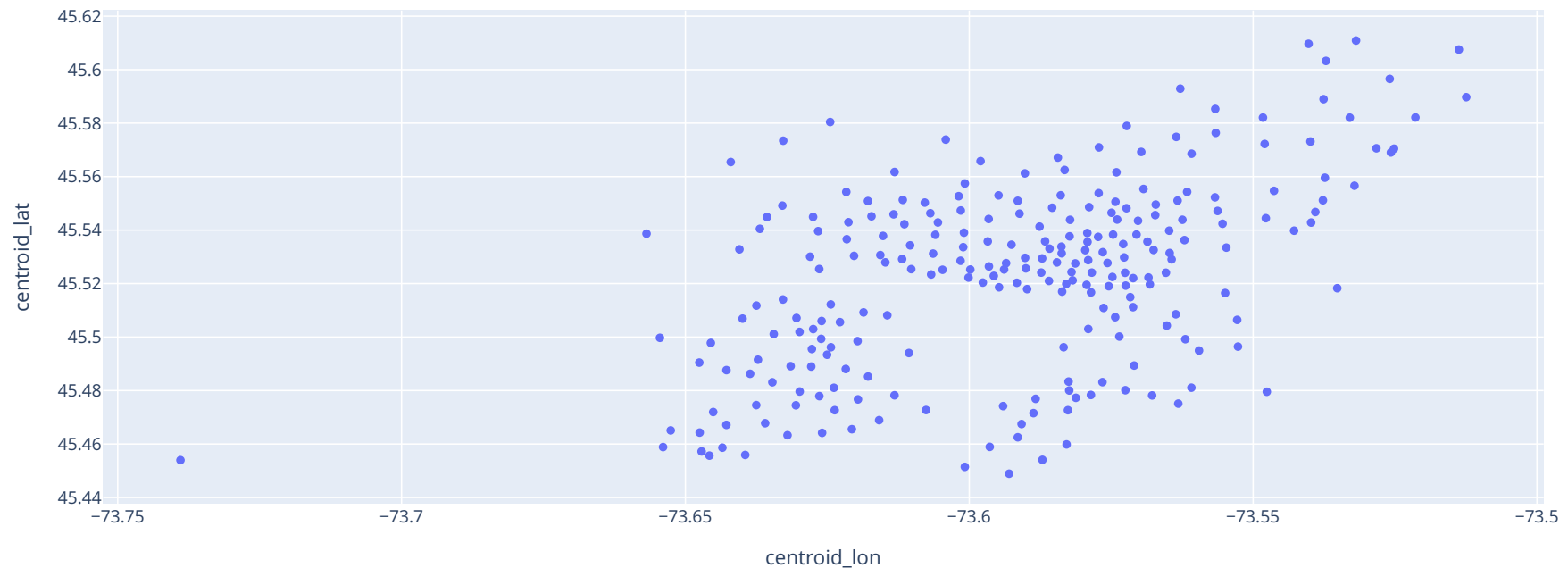
Next steps:

[Generate code with data\\_car](#)[View recommended plots](#)[New interactive sheet](#)

```
# Graficamos la ubicacion de los autos (lat y long)
px.scatter(data_frame=data_car, x='centroid_lon', y='centroid_lat', title='Ubicación de autos por coordenadas')
```

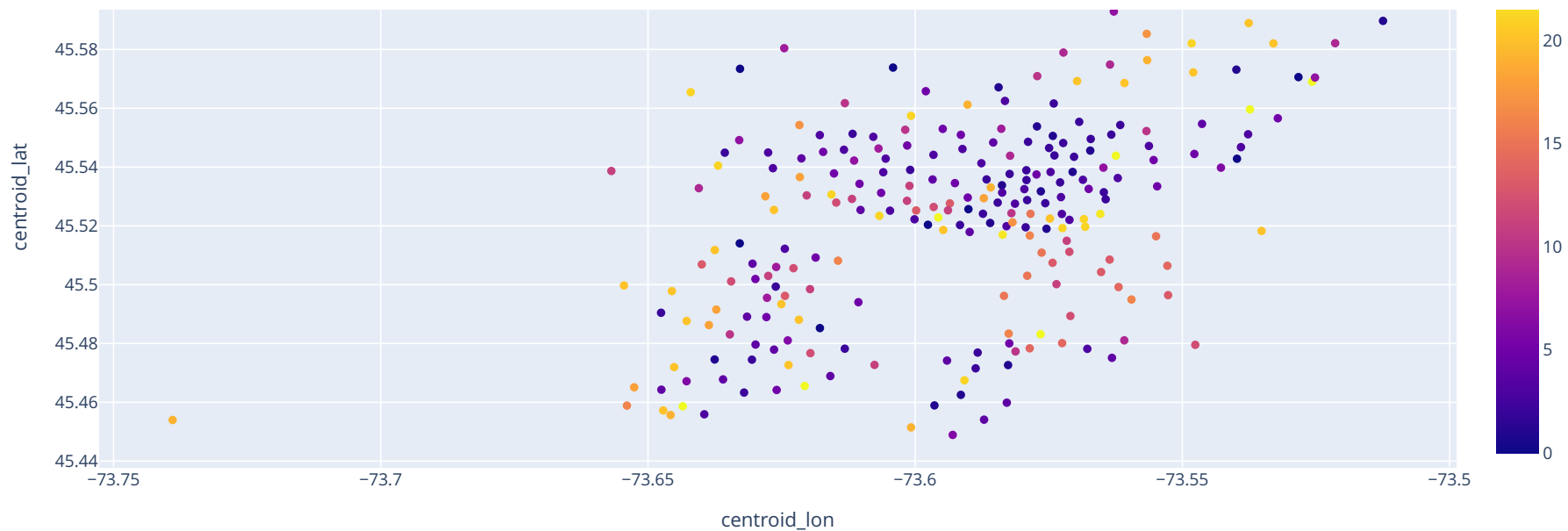


Ubicación de autos por coordenadas

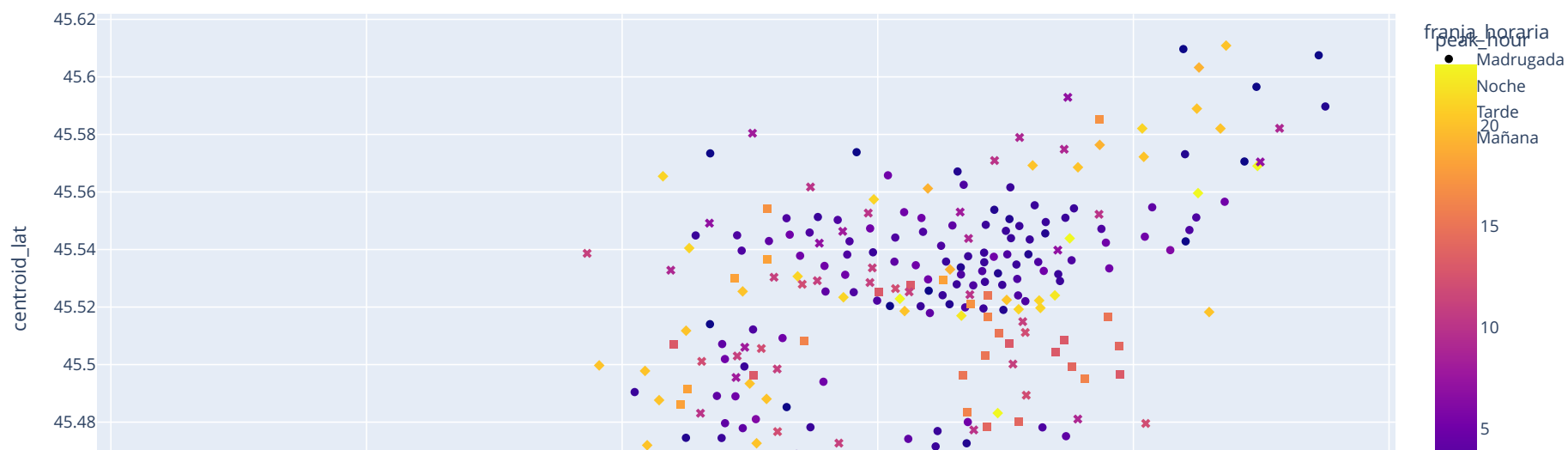


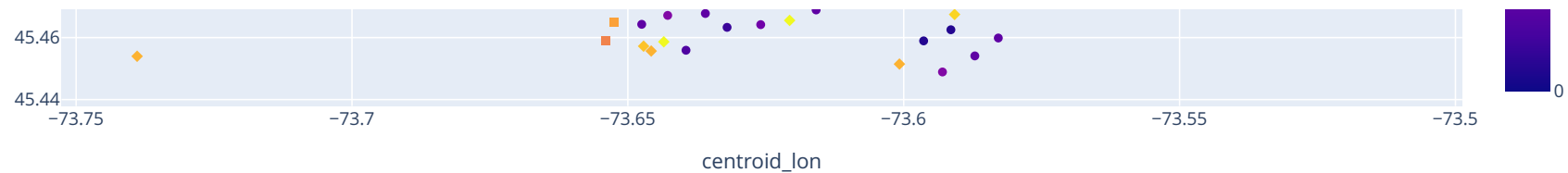
```
# Agregamos la configuracion 'color' en base a una de nuestras columnas
px.scatter(data_frame=data_car, x='centroid_lon', y='centroid_lat', color='peak_hour')
```





```
# Modificando la forma de los puntos
px.scatter(data_frame=data_car, x='centroid_lon', y='centroid_lat', color='peak_hour', symbol='franja_horaria')
```

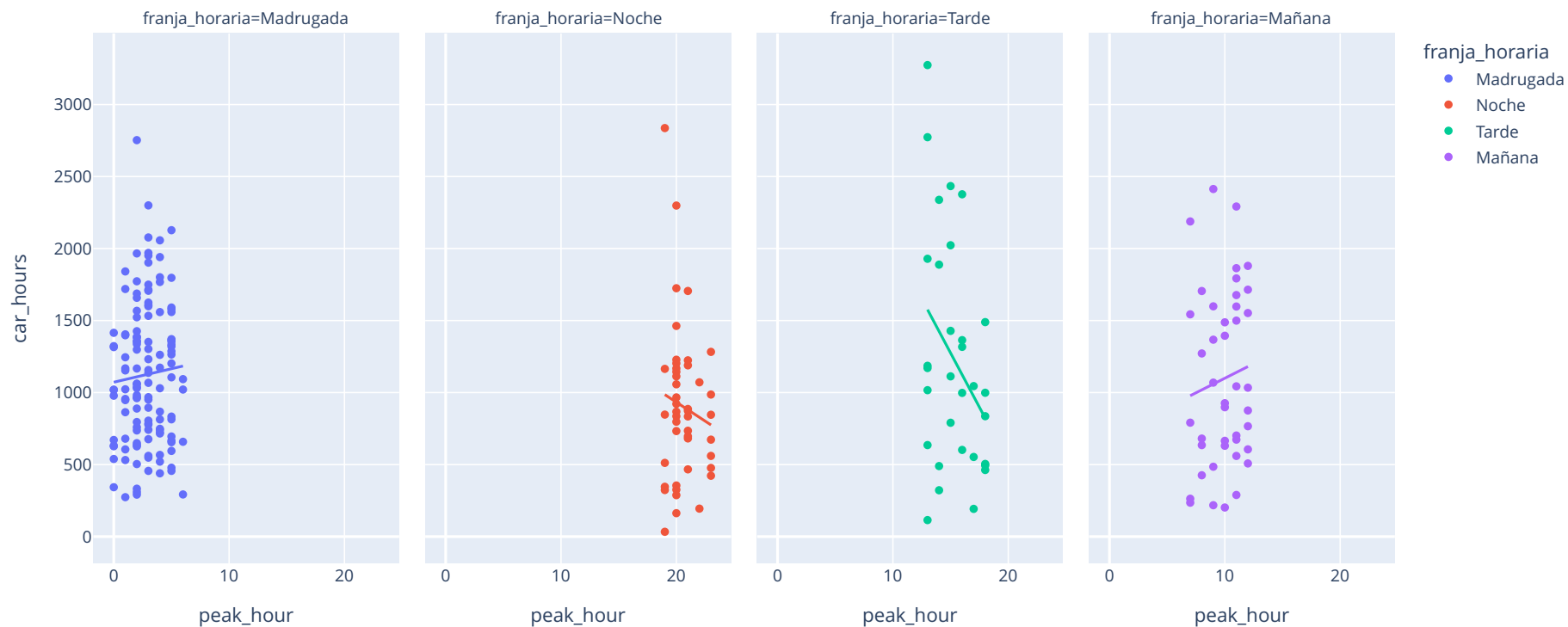




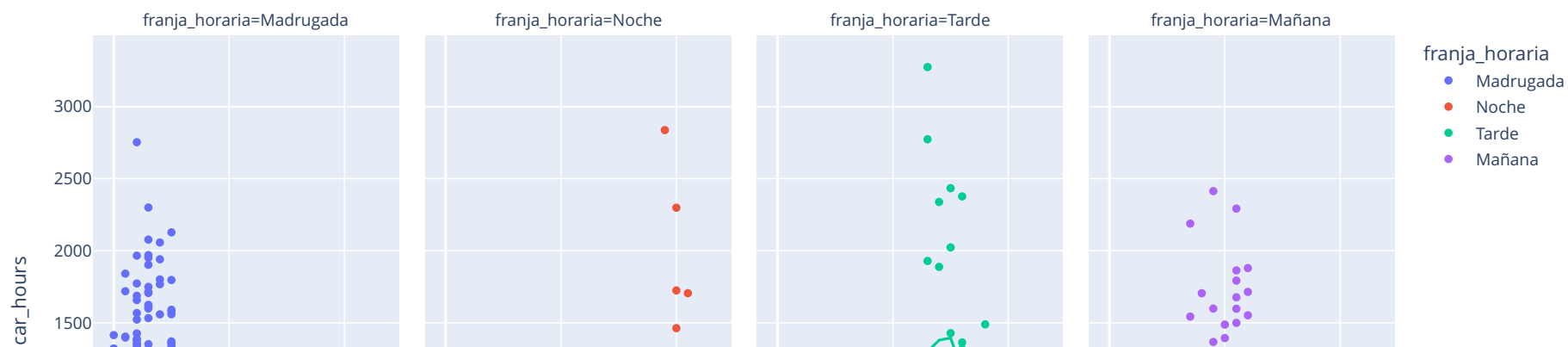
```
# Dvidiimos el grafico en diferentes subplots
px.scatter(data_frame=data_car, x='centroid_lon', y='centroid_lat', color='peak_hour', facet_col="franja_horaria")
```

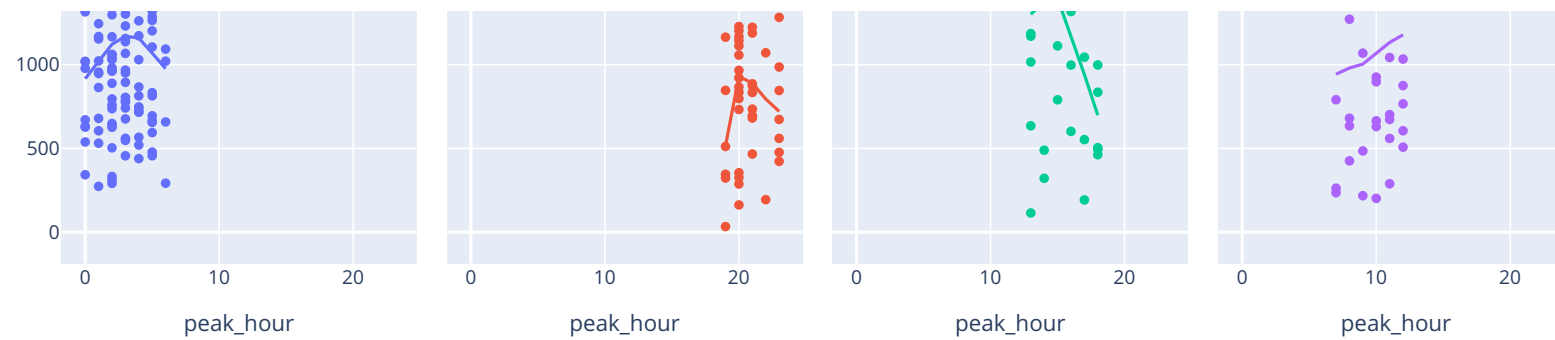


```
# Lineas de tendencia en subplots
px.scatter(data_frame=data_car, x='peak_hour', y='car_hours', color='franja_horaria', facet_col='franja_horaria', trendline='ols')
```

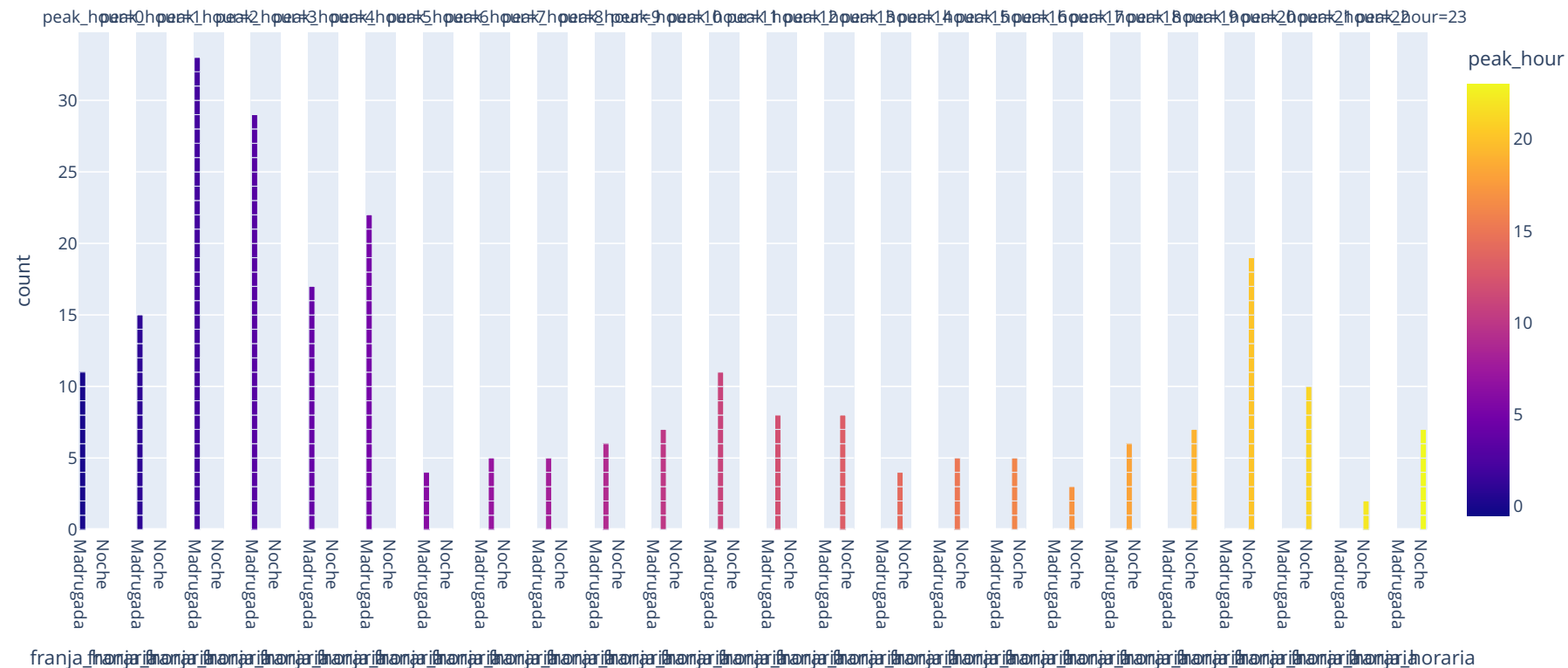


```
# Lineas de tendencia en subplots  
px.scatter(data_frame=data_car, x='peak_hour', y='car_hours', color='franja_horaria', facet_col='franja_horaria', trendline='lowess')
```

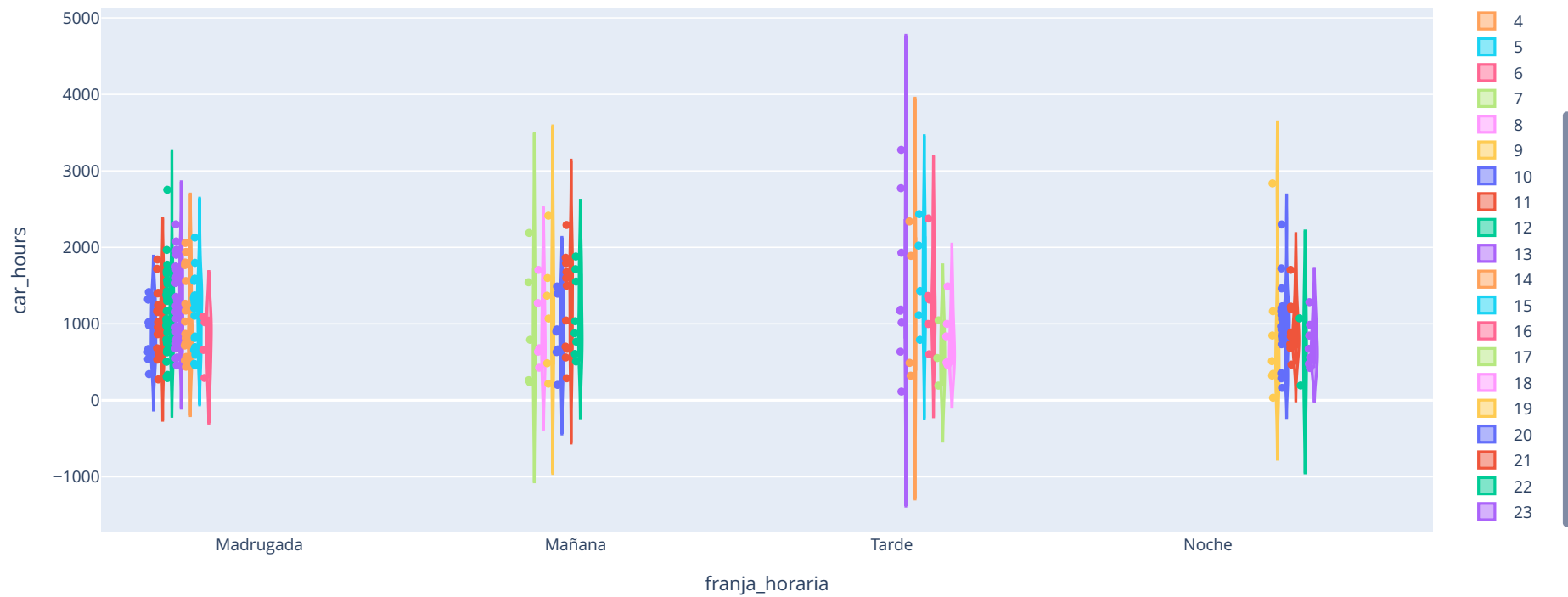




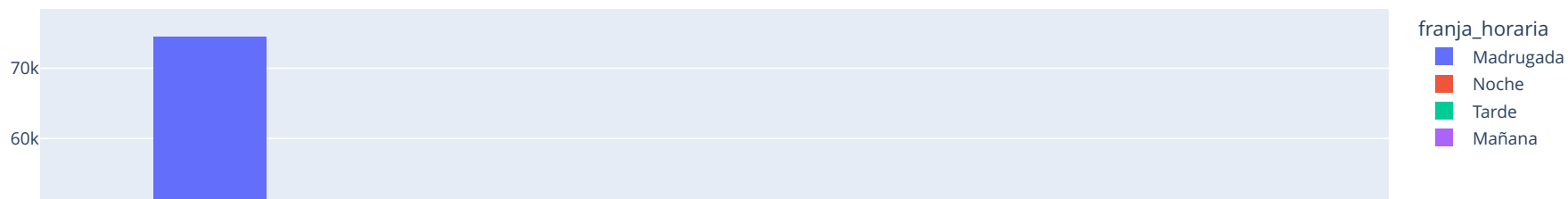
```
# Forma Analoga
px.bar(data_frame=data_car.sort_values(by='peak_hour'),x='franja_horaria',color='peak_hour',barmode='group',facet_col='peak_hour')
```

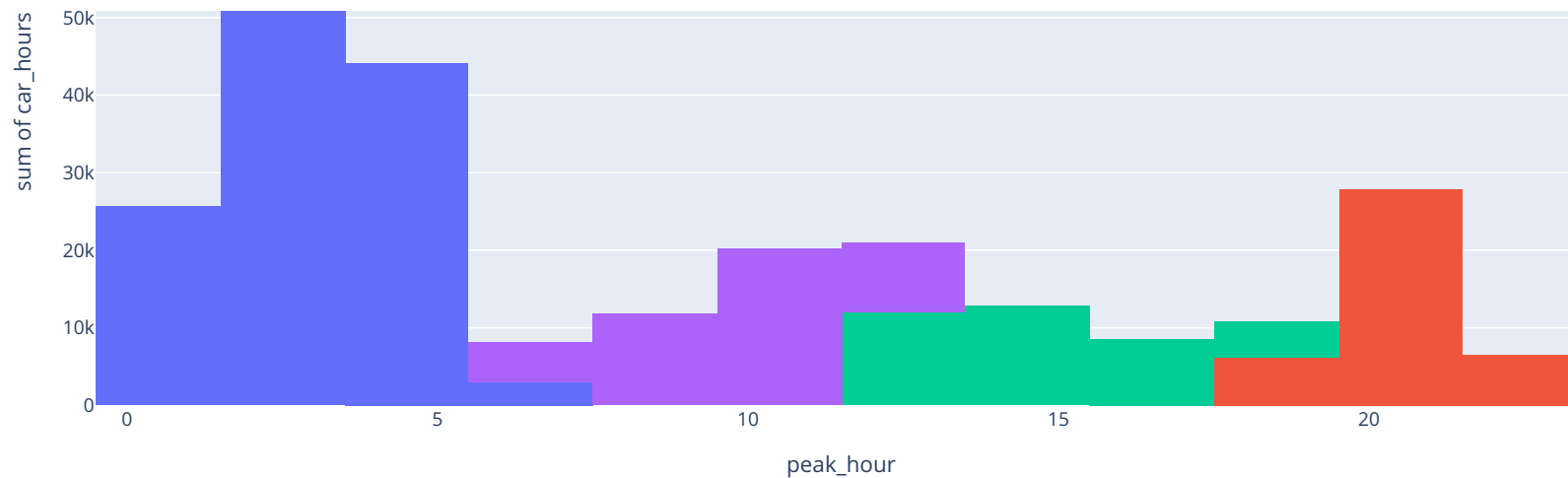


```
# Forma Analogia Violin
px.violin(data_frame=data_car.sort_values(by='peak_hour'), x='franja_horaria', y='car_hours', color='peak_hour', points='all', box=True)
```

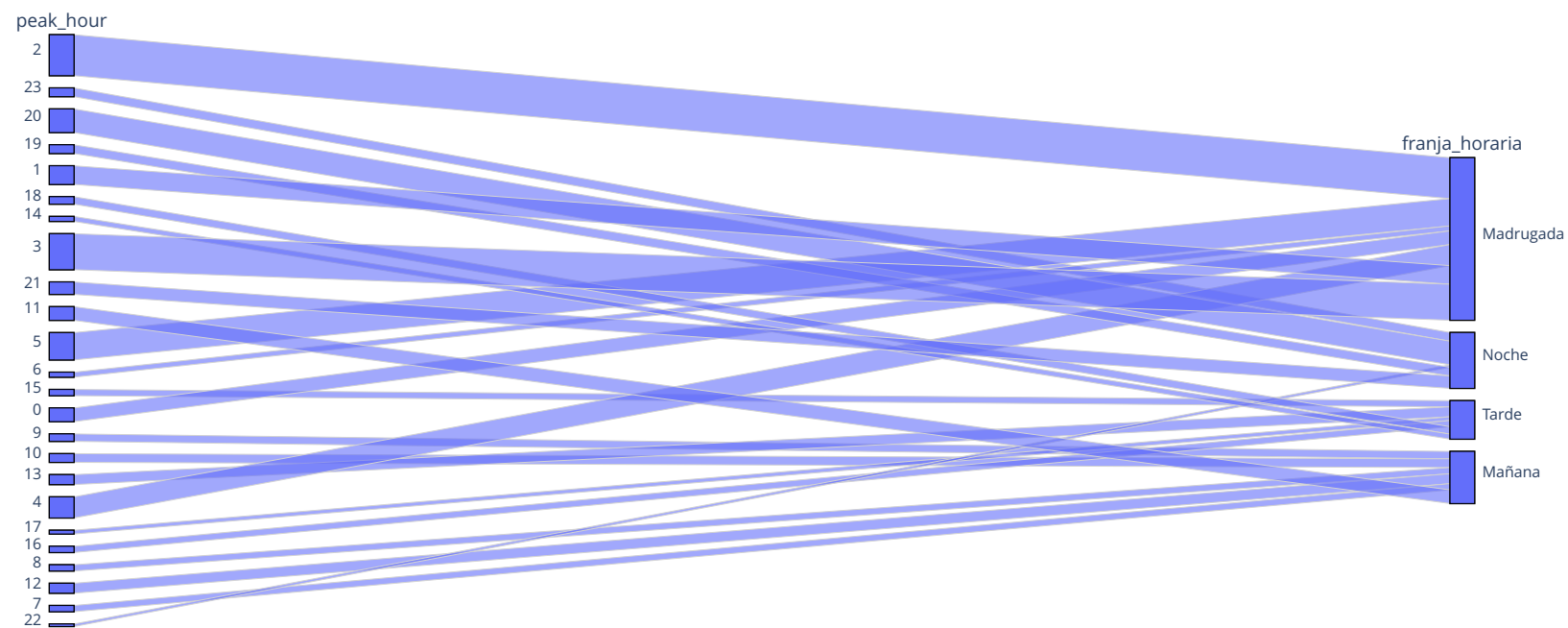


```
# Forma Analogia Histograma
px.histogram(data_frame=data_car, x='peak_hour', y='car_hours', color='franja_horaria')
```





```
# Forma Analoga Parallel Categories  
px.parallel_categories(data_frame=data_car, dimensions=['peak_hour', 'franja_horaria'])
```





```
# Forma Analoga Otras Variables de Color  
px.parallel_categories(data_frame=data_car,dimensions=['peak_hour', 'franja_horaria'],color='car_hours',color_continuous_scale=px.colors.s
```

