Smombies: Investigation of the nature and psychological implications of smartphone usage for entertainment

Mauro Vazquez Bassat (6718809), Malte Sichtermann (4441524)

1. Introduction

In today's fast-paced world, smartphones have become an integral part of our society, with more than two-thirds of the world's population owning one in 2023, according to recent statistics [1]. They are not just tools for communication but also serve as primary sources of entertainment. Whether it's scrolling through TikTok or Instagram, watching videos on YouTube, reading digital books, or following the latest sports news, our phones are always within reach. In fact, studies have shown that smartphones are often even used to influence and manage our emotional states, acting as coping mechanisms during stressful times or as a means to escape reality [2].

However, the impact of smartphone usage on our psychological well-being is complex and might impact our mental health over time [3]. This effect influenced by how we use our phones and how we perceive them and both positive and negative psychological outcomes have been observed, depending on these factors. Despite the growing body of research, there is still a significant gap in understanding the nature of these effects and the underlying reasons for their occurrence.

Our study aims to fill this gap by exploring this relationship between smartphone usage and emotional states and showing the extent to which the effects of smartphone use on the user differ depending on the context. We will leverage the K-EmoPhone dataset, a comprehensive collection of smartphone usage data from Korean students, which includes various measures of phone use and self-reported emotional states measured over a seven-day period [4]. It provides a valuable opportunity to study real-world smartphone usage patterns and their effects on users' emotions.

We are particularly interested in investigating how the combination of location and type of smartphone entertainment chosen by participants affects their emotional states. In order to gain comprehensive insights, we decided to distinguish between usage at home and usage on the go. This is to make the concept of location more comparable, as the dataset allows us to determine the participants' home location quite precisely. By analyzing this relationship, we hope to gain deeper insights into how different contexts and types of smartphone use affect users' emotional well-being differently. This is critical as it can inform better smartphone use habits and potentially guide the development of interventions to mitigate negative emotional outcomes associated with smartphone use. [5].

In the following sections, we will provide a detailed overview of the existing literature on smartphone usage and emotional states, describe the methodology used in our study, present our findings, and discuss their implications.

2. Related work

The use of smartphones for entertainment is an important area to investigate, as studies of the motivations behind mobile phone use have found the desire for entertainment, the need for information, and social interaction to be key drivers of smartphone use [6]. More recent studies that delve deeper into the effects of such entertainment-based uses, sometimes even refer to smartphones as tools for "instant happiness," used by individuals to manage their emotions, or providing immediate but temporary stress relief [2], or as "pacifying technologies" that alleviate feelings of anxiety and discomfort, offering users psychological relief in both social and solitary contexts [7]. Hoffner and Lee described this as the "dual nature" of smartphone use in emotion regulation back in 2015, noting that while phones provide immediate emotional support, their overuse could potentially lead to increased stress and reduced face-to-face interactions, which could impact long-term mental health [3]. Lepp raised similar concerns about the potential dangers of smartphone use on the quality of leisure time, such as fragmented attention and reduced physical activity due to the distraction and engagement provided by smartphones [8]. These concerns are further supported by a study showing how users tend to use their smartphones as a means of escapism and entertainment when engaging with them in leisure time [9], and the negative association between high frequency of mobile phone use and both academic performance and psychological well-being was again noted by Lepp in 2015 [10], when he examined their interplay with human character traits (introversion/extroversion). All of these studies suggest that the choice of different smartphone features for entertainment and the setting in which they are used could affect emotional levels and overall well-being of their users. While the existing research provides a solid foundation for understanding the dynamics of smartphone use and emotional impact, there remains a gap in exploring how big the impact of the environment is on these interactions and how the psychological impact differs based on the nature of the interaction. The K-EmoPhone dataset [4], offers a good opportunity to analyze those variables from a real-time data-set, therefore, we are confident that our study can contribute to the discourse on the psychological implications of smartphone use as a means of entertainment.

3. Methodology

3.1 Data Source

For this quantitative research study, all analyses were performed on data extracted from the K-EmoPhone dataset. The K-EmoPhone dataset is unique, combining phone and wearable wristband recordings with self-reported emotion, stress, and attention labels collected over a week from 77 Korean students. This allows for an in-depth investigation of the potential effects of smartphone usage, physical activity, and other measurements on the participants' emotional states.

3.2 Feature Selection

Given the extensive nature of the K-EmoPhone dataset, careful selection and combination of variables are crucial for the study's success. We focused on the real-time smartphone usage data, environmental conditions, and self-reported emotional and cognitive states to answer our research question effectively. For real-time smartphone usage data, we utilized the AppUsageEvent.csv file (Android smartphone) to determine how participants used their phones for leisure. For environmental conditions, data were sourced from both the smartwatch and smartphone, specifically from the Location.csv (Android smartphone), AmbientLight.csv (MS Band 2 smartwatch), and UltraViolet.csv (MS Band 2 smartwatch) files. Self-reported emotional and cognitive states were extracted from the EsmResponse.csv file, which contains responses to in-situ questionnaires completed by the participants.

3.3 Preprocessing and Feature Extraction

All preprocessing and feature extraction were conducted in a Jupyter Notebook environment using Python (see Appendix). The primary libraries used were pandas and os. The preprocessing steps included:
Data Loading and Initial Processing: Data was loaded into pandas DataFrames from CSV files for faster and simpler manipulation. Each participant's data was stored in individual DataFrames and later merged into a unique table per participant, which was stored in a dictionary for all participants.

Data Filtering: The AppUsageEvent.csv table was filtered to retain only leisure-related app usage. The Location.csv table had readings with an accuracy threshold of less than 20 meters removed.
Timestamp Handling: All tables with timestamp columns were converted to pandas datetime objects, localized to UTC, and then converted to Korean timezone. This ensured accurate merging using the merge_asof() method, which matches data points based on the nearest timestamp.
Home Coordinate Determination: To determine the participants' home coordinates, we analyzed significant time gaps in the ambient light data (although any smartwatch recordings data table would have also worked) and used these to identify consistent coordinate readings. Considering that the MS Band 2 required charging every night, and that there were no data recordings during charging, we identified the big differences in timestamp from one recording to the next and identified 6-7 clear points in time where the smartwatch was being charged for each participant (see Figure 1). The average longitude and latitude during these times were taken as the participant's home coordinates. A new feature, "at_home," was created using the haversine distance function to check if an event occurred within a 25-meter radius of the home coordinates.

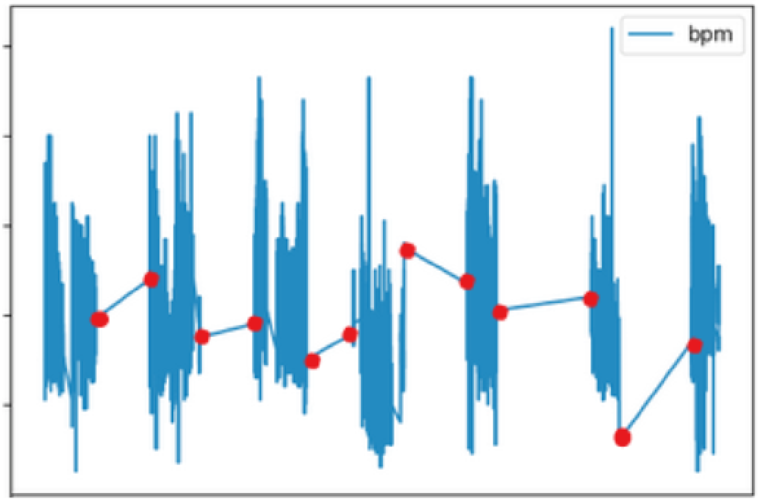


Figure 1: Intervals without data recordings

Feature Engineering: Categorical variables were transformed, and new features were created for analysis. Emotional and cognitive states were encoded into binary variables (e.g., stressed vs. not stressed), and brightness levels were categorized into low, medium, and high.
Determining Entertainment App Categories: The AppUsageEvent table's "category" field values were retrieved from Google Play or application archive websites. Categories deemed irrelevant (e.g., PERSONALIZATION, COMMUNICATION) were excluded. Relevant categories included VIDEO_PLAYERS, MUSIC_AND_AUDIO, SOCIAL, GAME, SHOPPING, NEWS_AND_MAGAZINES, SPORTS, BOOKS_AND_REFERENCE, COMICS. Some apps were manually reclassified for better accuracy.
Data Integration: Preprocessed tables were merged into a single dataset based on participant numbers and timestamps, ensuring a comprehensive dataset.
Saving Preprocessed Data: The final preprocessed dataset was saved for subsequent statistical analysis (see Figure 2).

timestamp	name	category	brightness	at_home	stress	valence	arousal	attention	pcode
2019-05-04 21:33:06.329000+09:00	네이버 웹툰	COMICS	LOW	False	0.0	0.0	0.0	0.0	P69
2019-05-21 12:26:44.090000+09:00	YouTube	VIDEO_PLAYERS	MEDIUM	False	0.0	1.0	0.0	1.0	P50
2019-05-04 19:24:58.598000+09:00	Pokémon GO	GAME	LOW	False	0.0	1.0	1.0	1.0	P64
2019-05-01 12:47:31.174000+09:00	Instagram	SOCIAL	LOW	False	0.0	1.0	1.0	1.0	P79
2019-05-11 02:36:27.671000+09:00	Facebook	SOCIAL	LOW	False	0.0	1.0	0.0	0.0	P17

Figure 2: Excerpt of the preprocessed data

3.4 Statistical Analysis

Statistical analysis was conducted in a Jupyter Notebook using Python (see Appendix). Key libraries included pandas, statsmodels, seaborn, and matplotlib. The analysis involved descriptive statistics and logistic regression models to understand the relationships between smartphone usage, physical activity, and emotional states.

Descriptive Statistics: Descriptive statistics provided an overview of the dataset, summarizing key characteristics such as the distribution of app categories, brightness levels, and the proportion of time participants spent at home. The descriptive analysis helped identify the central tendencies, variability, and overall patterns within the data. For example, visualizations were created to show the distribution of app usage across different categories and how these varied when participants were at home versus away. The analysis also highlighted the prevalence of various emotional states (stress, valence, arousal, attention) within the participant group.

Logistic Regressions with Fixed Effects: Four different logistic regression models were employed to evaluate the influence of various predictors on binary outcomes related to emotional and cognitive states (stress, valence, arousal and attention). The analysis included both main and interaction effects while controlling for individual differences among participants.

Main Effects: The models assessed how app category, and being at home influenced stress, valence, arousal, and attention.

Interaction Effects: The models evaluated whether the impact of type of app usage on emotional states differed based on location (at home vs. elsewhere).

Individual Differences: By controlling for unique characteristics of each participant, the models aimed to enhance the accuracy of the results. Logistic regression was chosen because it is well-suited for modeling binary outcomes, such as whether a person is stressed (yes/no), feeling good (yes/no), attentive (yes/no), or aroused (yes/no). This method allows for estimating the probability of these outcomes based on predictors (app category, location, brightness) and their interactions. Additionally, logistic regression accounts for individual differences through fixed effects, providing a more nuanced understanding of the relationships between predictors and outcomes.

4. Results

4.1 Descriptive Statistics

See Figure 3 for a more detailed overview.

4.1.1 Count and Unique Values

The final processed data table used for the analysis had a total of 189,861 records. The data included activity from all 77 participants, although the distribution was uneven, as not all participants used their phones for entertainment equally (see Figure 4). A total of 267 different app names were recorded across 10 different entertainment app categories, with Facebook being the most frequent.

	timestamp	name	category	brightness	at_home	stress	valence	arousal	attention	pcode
count	189861	189861	189861	189861	189861	189861.000000	189861.000000	189861.000000	189861.000000	189861
unique	189843	267	10	3	2	NaN	NaN	NaN	NaN	77
top	2019-05-16 23:34:27.647000+09:00	Facebook	SOCIAL	LOW	False	NaN	NaN	NaN	NaN	P56
freq	2	46992	107528	177576	152343	NaN	NaN	NaN	NaN	11036
mean	NaN	NaN	NaN	NaN	NaN	0.312797	0.549628	0.366310	0.472014	NaN
std	NaN	NaN	NaN	NaN	NaN	0.463634	0.497532	0.481797	0.499217	NaN
min	NaN	NaN	NaN	NaN	NaN	0.000000	0.000000	0.000000	0.000000	NaN
25%	NaN	NaN	NaN	NaN	NaN	0.000000	0.000000	0.000000	0.000000	NaN
50%	NaN	NaN	NaN	NaN	NaN	0.000000	1.000000	0.000000	0.000000	NaN
75%	NaN	NaN	NaN	NaN	NaN	1.000000	1.000000	1.000000	1.000000	NaN
max	NaN	NaN	NaN	NaN	NaN	1.000000	1.000000	1.000000	1.000000	NaN

Figure 3: Overview of the descriptive statistics

4.1.2 Data Distribution

Participants

Most participants used their phones for entertainment purposes at low to moderate levels. A smaller group of participants displayed moderate usage, while a few outliers exhibited very high usage.

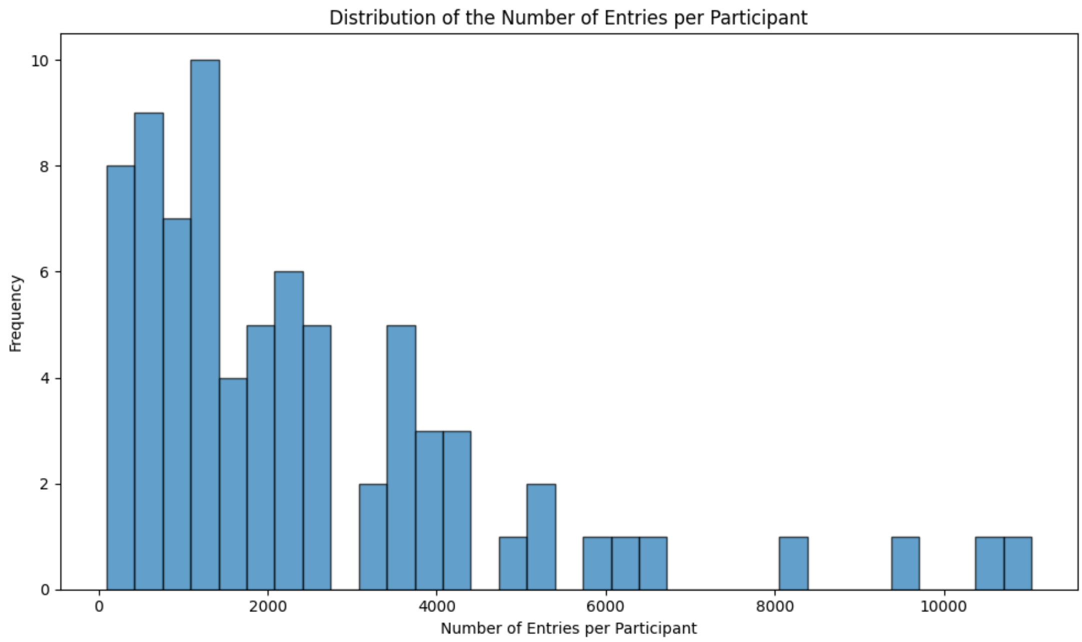


Figure 4: Distribution of phone usage for entertainment purposes per participant

At Home

The distribution of records was unbalanced, with 152,343 records not at home and 37,518 at home. This imbalance might be due to no data being recorded from 10 pm until the morning, a time when many people use their phones before bed or late at night.

Category

As expected, social media was the most frequent app category by far, with over 100,000 records. This is consistent with Facebook being the most frequently used app. Social media apps often also serve as communication apps. See the app category distributions in Figure 5. Additionally, see the proportion of different app categories used by participants when they were at home compared to when they were not at home in Figure 6.

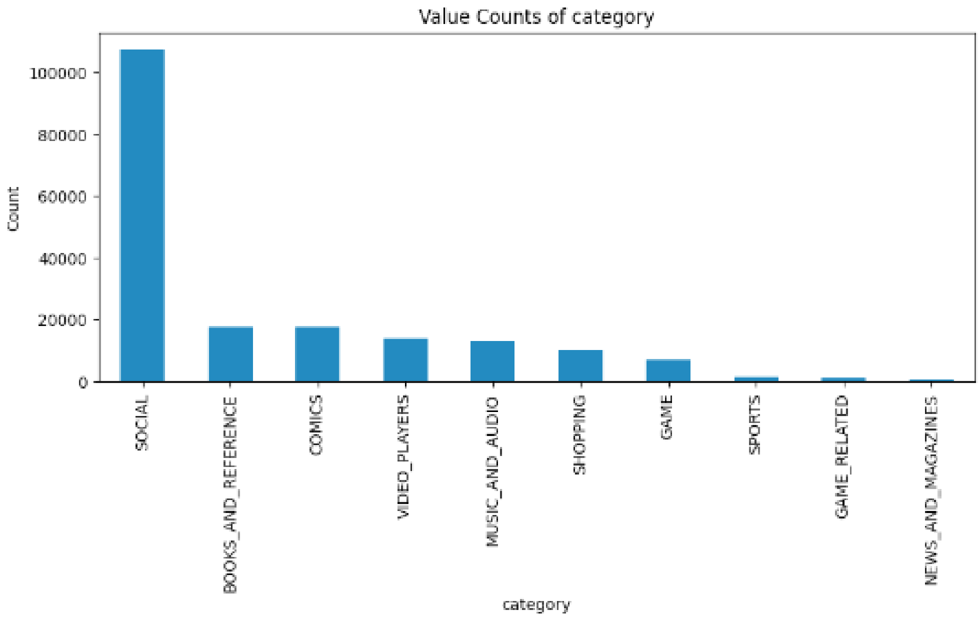


Figure 5: Distribution of phone usage data by app category

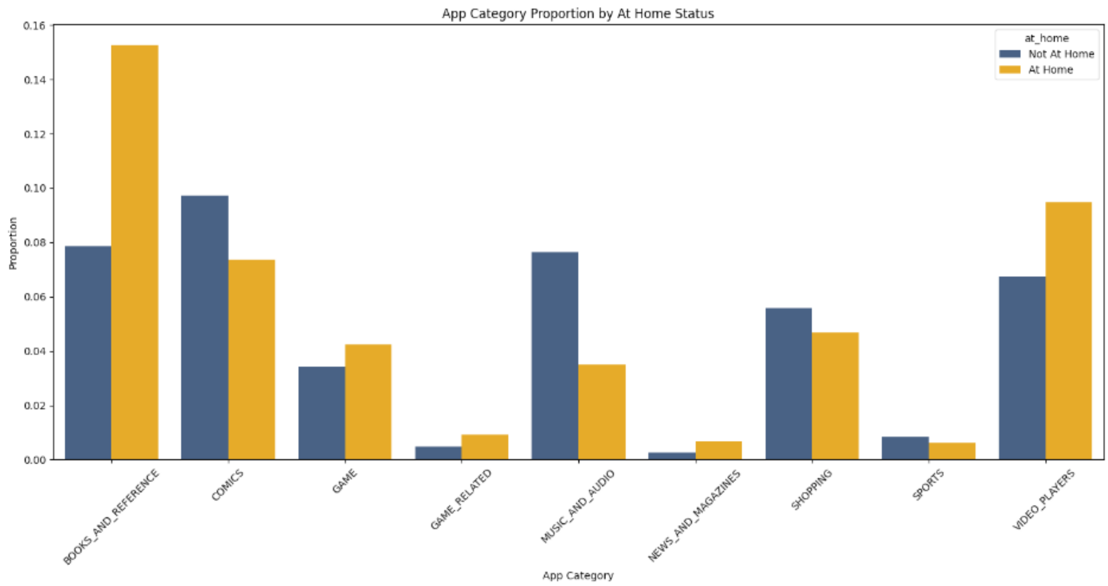


Figure 6: Distribution of phone usage data by app category **at home** vs **outdoors**

Brightness Distribution

The brightness distribution in our dataset was very unbalanced, causing issues for the logistic regression model. Consequently, we decided to remove this variable from our analysis. The imbalance might be due to factors such as the sensor being covered by sleeves (see Appendix L).

4.2 Logistic Regression with Fixed Effects

For full summary tables of the models see Appendixes T, U, V, W, X, and Y.

4.2.1 Stress

The logistic regression model for stress revealed several statistically significant predictors:

- **Game Related Apps**: Positive association with stress ($\beta = 1.1210$, $p < 0.001$).

- **Music and Audio Apps**: Positive association with stress ($\beta = 0.4491$, $p < 0.001$).
- **News and Magazines**: Positive association with stress ($\beta = 1.0659$, $p < 0.001$).
- **Social Apps**: Positive association with stress ($\beta = 0.2109$, $p < 0.001$).
- **Sports Apps**: Positive association with stress ($\beta = 0.4265$, $p < 0.001$).
- **Video Players**: Positive association with stress ($\beta = 0.3194$, $p < 0.001$).
- **At Home**: Being at home is positively associated with stress ($\beta = 0.2775$, $p < 0.001$).
- **Interaction Effects**: Significant negative interactions were observed for several categories when at home, such as game-related apps ($\beta = -1.3480$, $p < 0.001$) and music and audio apps ($\beta = -1.0196$, $p < 0.001$).

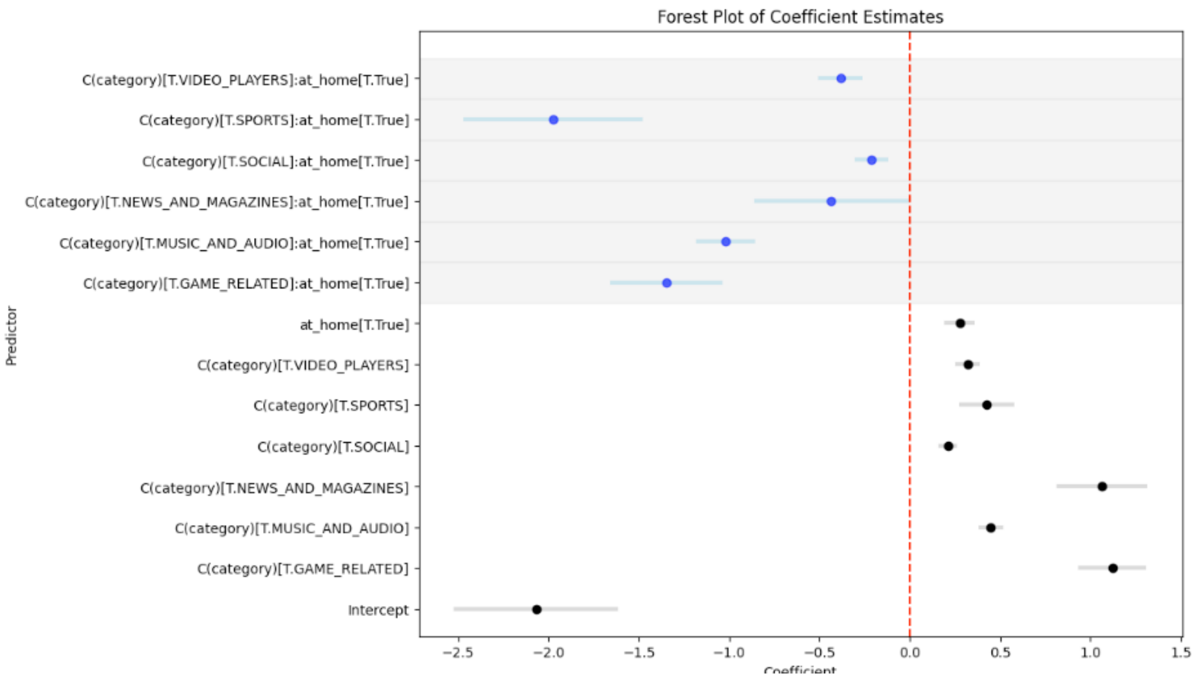


Figure 7: Coefficient estimates for **stress**

4.2.2 Valence

The logistic regression model for valence identified the following significant predictors:

- **Comics**: Positive association with valence ($\beta = 0.1076$, $p = 0.003$).
- **Game Related Apps**: Negative association with valence ($\beta = -0.3977$, $p < 0.001$).
- **Music and Audio Apps**: Negative association with valence ($\beta = -0.1374$, $p < 0.001$).
- **Shopping**: Positive association with valence ($\beta = 0.0968$, $p = 0.025$).
- **Social Apps**: Negative association with valence ($\beta = -0.1253$, $p < 0.001$).
- **Sports Apps**: Negative association with valence ($\beta = -0.6644$, $p < 0.001$).
- **At Home**: Negative association with valence ($\beta = -0.1763$, $p < 0.001$).
- **Interaction Effects**: Significant positive interactions were observed for game-related apps ($\beta = 1.4006$, $p < 0.001$) and music and audio apps ($\beta = 0.9336$, $p < 0.001$) when at home.

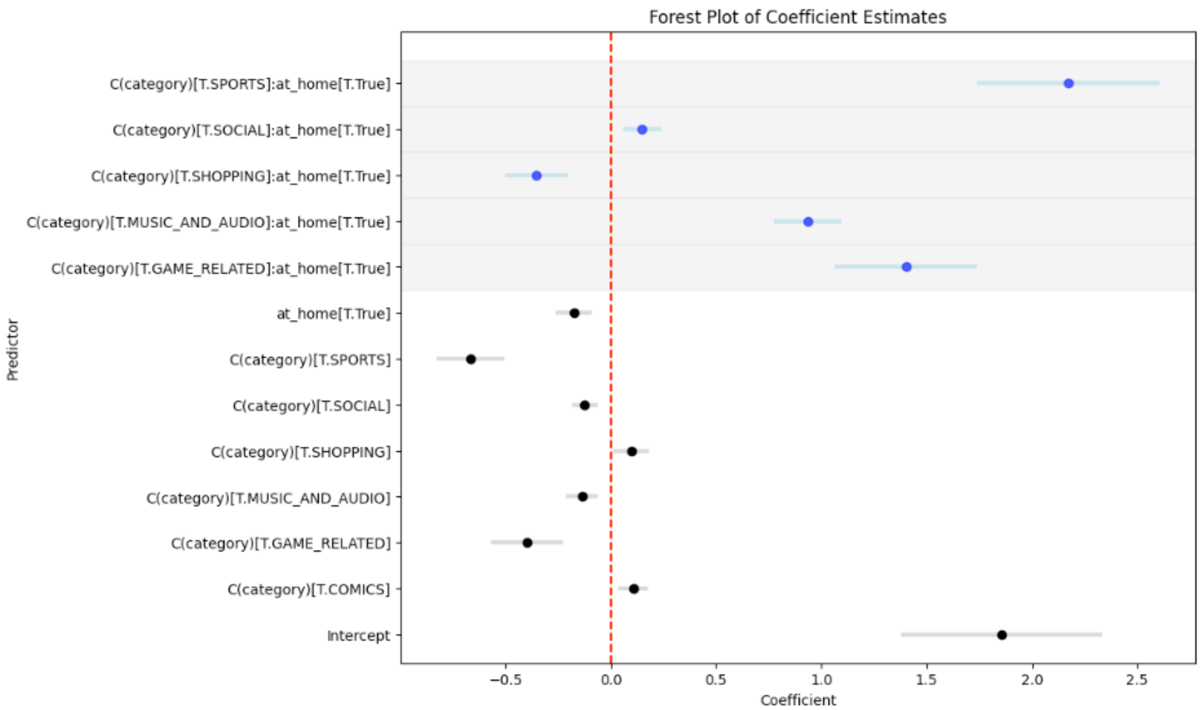


Figure 8: Coefficient estimates for **valence**

4.2.3 Arousal

The logistic regression model for arousal highlighted the following significant predictors:

- **Comics**: Positive association with arousal ($\beta = 0.2706$, $p < 0.001$).
- **Game Related Apps**: Negative association with arousal ($\beta = -0.3670$, $p = 0.004$).
- **Music and Audio Apps**: Negative association with arousal ($\beta = -0.2522$, $p < 0.001$).
- **Shopping**: Negative association with arousal ($\beta = -0.4937$, $p < 0.001$).
- **Sports Apps**: Negative association with arousal ($\beta = -0.3977$, $p < 0.001$).
- **Video Players**: Positive association with arousal ($\beta = 0.1180$, $p = 0.001$).
- **At Home**: Negative association with arousal ($\beta = -0.2210$, $p < 0.001$).
- **Interaction Effects**: Significant positive interactions were observed for game-related apps ($\beta = 1.1134$, $p < 0.001$) and music and audio apps ($\beta = 0.5517$, $p < 0.001$) when at home.

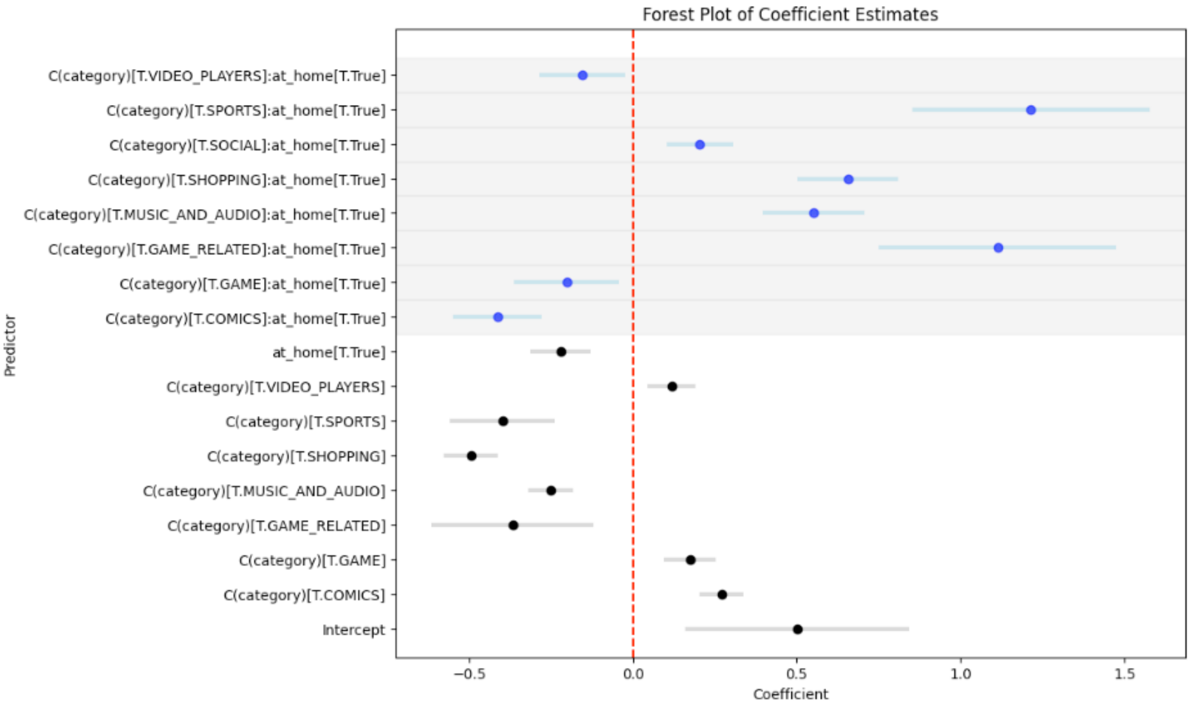


Figure 9: Coefficient estimates for **arousal**

4.2.4 Attention

The logistic regression model for attention revealed the following significant predictors:

- **Comics**: Positive association with attention ($\beta = 0.2660$, $p < 0.001$).
- **Game Related Apps**: Negative association with attention ($\beta = -0.1860$, $p < 0.001$).
- **Music and Audio Apps**: Negative association with attention ($\beta = -0.1159$, $p = 0.001$).
- **News and Magazines**: Negative association with attention ($\beta = -0.3765$, $p = 0.003$).
- **Shopping**: Negative association with attention ($\beta = -0.0944$, $p = 0.016$).
- **Sports Apps**: Negative association with attention ($\beta = -0.3566$, $p < 0.001$).
- **At Home**: Negative association with attention ($\beta = -0.2210$, $p < 0.001$).
- **Interaction Effects**: Significant positive interactions were observed for music and audio apps ($\beta = 1.0083$, $p < 0.001$) and news and magazines ($\beta = 1.1877$, $p < 0.001$) when at home.

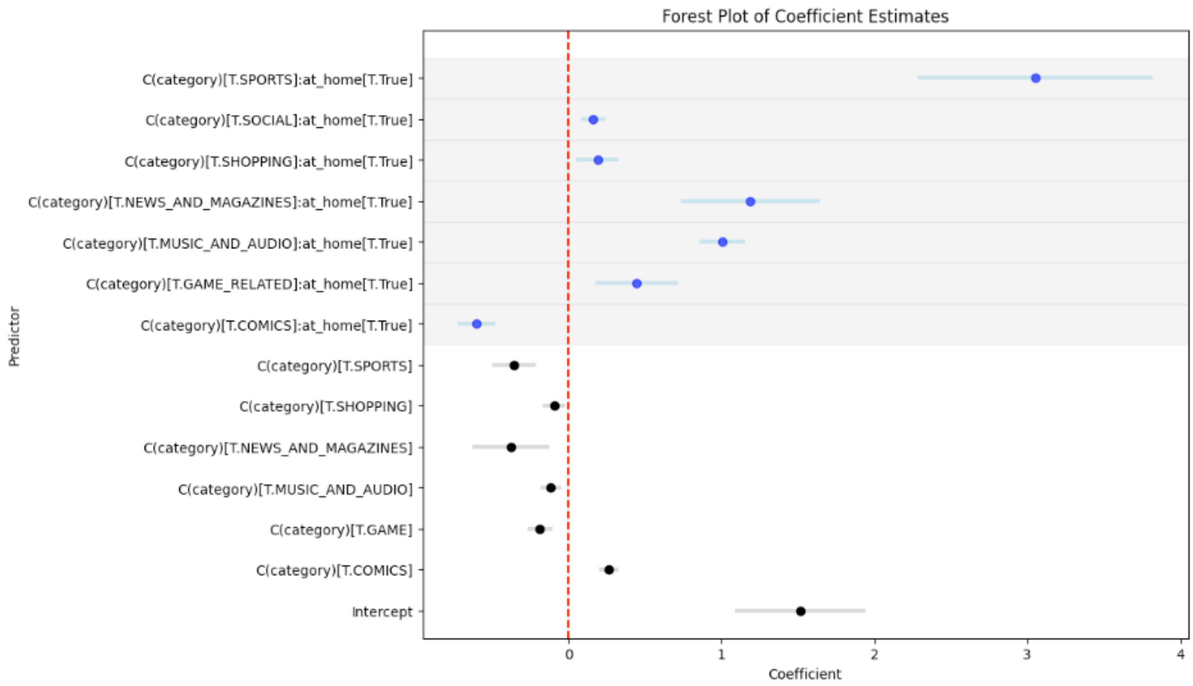


Figure 10: Coefficient estimates for **attention**

5. Discussion and conclusion

Our study aimed to explore how different contexts and types of smartphone use affect users' emotional well-being and the logistic regressions with fixed effects yielded several interesting results. As this analysis method by nature controls for the variables specified as fixed effects, and provides direct estimates of the relationships we examine, we did not perform additional post-hoc analysis [11]. Instead, we will provide an explanation and interpretation of the results below.

5.1 Main Effects

Stress: Apps from the categories "GAME_RELATED" ($\beta = 1.1210, p < 0.001$), "MUSIC_AND_AUDIO" ($\beta = 0.4491, p < 0.001$), "NEWS_AND_MAGAZINES" ($\beta = 1.0659, p < 0.001$), and "SOCIAL" ($\beta = 0.2109, p < 0.001$) were associated with higher stress levels, with game-related apps and news and magazine apps showing the highest effects, while the effect of using social apps was the smallest. These results are surprising, as social apps are often discussed as a source of stress due to social comparison and the pressure to constantly interact with others. One possible explanation for the lower impact is that the social aspect of connecting with friends compensates for these factors. For game-related apps, the impact could be explained by their often competitive nature, and for news and magazines, the reason could be the often negative or overwhelming content they provide. Music and audio apps as well as sports apps ($\beta = 0.4265, p < 0.001$) also show a positive association with stress, possibly due to highly stimulating or intense content. Same goes for Video players ($\beta = 0.3194, p < 0.001$), but their effect is less pronounced, probably because the content is less fast-paced than in games or social media and there is less social pressure.

Interestingly, being at home is also positively associated with stress ($\beta = 0.2775, p < 0.001$), although the effect is rather small. This observation might be explained by the context, that "at home" in our study always refers to "at home and using apps for entertainment", which might often happen in a context when users are already stressed and use these apps to mitigate it. Looking at the highly significant negative interaction effects that were observed for several categories of apps when being at home, such as game-related apps ($\beta = -1.3480, p < 0.001$) and music and audio apps ($\beta = -1.0196, p < 0.001$) adds to that suggestion. While these apps generally increase stress, their effects seem to be reduced or even reversed when the person is at home. This suggests that the effects of using these apps does indeed depend on the environmental context.

Valence: Apps from the categories "GAME_RELATED" ($\beta = -0.3977, p < 0.001$), "MUSIC_AND_AUDIO" ($\beta = -0.1374, p < 0.001$), as well as "SOCIAL" ($\beta = -0.1253, p < 0.001$) were associated with lower valence, indicating a negative impact on positive emotional states. As with stress, Game-related apps showed the highest effects, while the effect of social apps was the smallest, likely for similar reasons as the ones mentioned above. For music and audio apps, the impact could be explained by the potentially overwhelming or mood-altering effects they provide. Interestingly, comics ($\beta = 0.1076, p = 0.003$) and shopping apps ($\beta = 0.0968, p = 0.025$) show a positive association with valence. For comics that might be because they provide entertainment and relaxation, leading to an improved mood, and shopping apps might elevate valence by offering a sense of pleasure or reward through browsing and purchasing activities.

Being at home was also negatively associated with valence ($\beta = -0.1763, p < 0.001$), although the effect is rather small and this might again be explained by the context under that we describe "at home". Once again, there were some highly significant positive interaction effects for game-related apps ($\beta = 1.4006, p < 0.001$) and music and audio apps ($\beta = 0.9336, p < 0.001$), adding to the conjecture, that using these apps at home seems to have a more beneficial effect on the emotional states than using them elsewhere.

Arousal: Only apps from the categories "COMICS" ($\beta = 0.2706, p < 0.001$) and "VIDEO PLAYERS" ($\beta = 0.1180, p = 0.001$) were positively associated with arousal, indicating that individuals feel more stimulated when engaging with comic-related or long-form video content. Once again, "GAME_RELATED" apps ($\beta = -0.3670, p = 0.004$) exhibited a negative association with arousal which is surprising because one would rather expect an increase of arousal when playing mobile games. This might be explained by the oftentimes repetitive gameplay in base-builder or idle-games, which make up for a big share of mobile games nowadays. "MUSIC_AND_AUDIO" apps ($\beta = -0.2522, p < 0.001$) also showed a negative association with arousal, which in this case is consistent with the expectation of them being used for relaxation and mood regulation. "SHOPPING" apps ($\beta = -0.4937, p < 0.001$) were strongly negatively associated with arousal, likely due to the nature of online shopping which while rewarding at first can quickly become disappointing. Similarly, "SPORTS" apps ($\beta = -0.3977, p < 0.001$) had a negative association with arousal, possibly because following sports news via e.g. live-tickers might not be as engaging as watching live events.

Being at home was also negatively associated with arousal ($\beta = -0.2210, p < 0.001$), in this case expected, because the home environment, typically associated with relaxation, might reduce arousal during smartphone use. Again, significant interaction effects were found for several categories of apps when used at home, such as "GAME_RELATED" apps ($\beta = 1.1134, p < 0.001$) and "MUSIC_AND_AUDIO" apps ($\beta = 0.5517, p < 0.001$), likely connected to the intentions and expectations of using these apps in a home setting.

Attention: Apps from the category "COMICS" ($\beta = 0.2660, p < 0.001$) were associated with higher attention levels, suggesting that engaging with comics might enhance focus and concentration, possibly due to their structured and narrative-driven content. In contrast, "GAME_RELATED" apps ($\beta = -0.1860, p < 0.001$), "MUSIC_AND_AUDIO" apps ($\beta = -0.1159, p = 0.001$), "NEWS_AND_MAGAZINES" ($\beta = -0.3765, p = 0.003$), "SHOPPING" ($\beta = -0.0944, p = 0.016$), and "SPORTS" apps ($\beta = -0.3566, p < 0.001$) were negatively associated with attention. The negative association with game-related apps might be due to their highly stimulating and often distracting nature (e.g. when using them while being supposed to do something else). Similarly, music and audio apps may contribute to background noise that hampers focus, while news and magazines can be overwhelming or distressing, taking attention away from other tasks. Shopping apps involve browsing and decision-making processes that can increase decision fatigue, and sports apps often provide frequent updates and notifications that disrupt attention. All apps from these categories are usually not used to improve focus (as their main purpose is entertainment) therefore these results are not surprising.

Interestingly though, being at home was also negatively associated with attention ($\beta = -0.2210, p < 0.001$), suggesting that the home environment might not be beneficial to maintaining attention, likely because it is easy to find various distractions. However, significant positive interaction effects were observed for "MUSIC_AND_AUDIO" apps ($\beta = 1.0083, p < 0.001$) and "NEWS_AND_MAGAZINES" ($\beta = 1.1877, p < 0.001$) when used at home which also seems intuitive. Individuals might for example use music and audio apps to create a focused environment when learning, or they may be able engage with news and magazines in a comfortable setting better.

5.2 Validity and Limitations

Our findings align with existing literature on the dual nature of smartphone use in emotion regulation [3] by showing the effects interacting with the smartphone has on its user. In addition, they supplement existing research by breaking down what causes these effects and which factors influence them more or less strongly (positively or negatively). However, the study does have some limitations. The K-Emophone dataset is limited to Korean students, which may make generalizing findings to other populations difficult. Additionally, the self-reported nature of emotional states can introduce bias and there was a slight offset between the recording times of the in-situ questionnaires and the time records of the wearable & smartphone data. We tried to account for that by using the nearest timestamps, but it might still cause some inaccuracy. Another strong limitation of the dataset is the lack of recordings during night, as the study was only conducted between 10AM and 10PM. This might especially influence insights into the "at home" data, because many uses of the phone probably occur before or after this interval. Last but not least, the way we determined participants' home coordinates might lead to some inaccuracies, for example if participants weren't at home when they were sleeping or if the determined area did not precisely match their home.

5.3 Future Work

Our work is intended to draw attention to the extent to which the effects of smartphone use on the user differ depending on the context. The results show that there are indeed strong differences in the effects depending on whether the user is at home or outside and also depending on which form of entertainment he chooses in the given situation. The significant findings reported can provide a good starting point for further research into these (interactive) effects and may help to narrow down which areas are interesting and which are less relevant. Longitudinal studies to assess long-term effects of the mentioned interactions and more diverse samples for better generalisability would enhance the body of research on this area. Also exploring other contextual factors, such as social interactions and physical environment, could be an interesting step to provide a more comprehensive understanding of smartphone usage's impact on emotional well-being.

5.4 Conclusion

In this study, we set out to understand how smartphone usage and environmental context impact users' emotions. Our findings confirm that the type of app used and whether the user is at home or elsewhere significantly influence emotional states. By using the real-time data from the K-EmoPhone dataset and controlling for individual differences, we provided clear evidence of these effects. This research is a starting point to filling the gaps identified in the introduction, showing that context and app choice are critical in determining the emotional outcomes of smartphone use. These insights can help guide healthier smartphone habits and inform interventions to reduce negative impacts, ultimately contributing to better mental health and well-being.

References

[1] F. Laricchia, *Smartphone penetration worldwide as share of global population 2016-2023*, Last visited 13.06.2024, May 2024. [Online]. Available: https://www.statista.com/statistics/203734/global-smartphone-penetration-per-capita-since-2005/.

[2] Y. Shi, P. Koval, V. Kostakos, J. Goncalves and G. Wadley, '"Instant Happiness": Smartphones as tools for everyday emotion regulation,' *International Journal of Human-Computer Studies*, vol. 170, p. 102 958, Feb. 2023, ISSN: 1071-5819. DOI: 10.1016/j.ijhcs.2022.102958. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1071581922001732 (visited on 13/05/2024).

[3] C. A. Hoffner and S. Lee, 'Mobile Phone Use, Emotion Regulation, and Well-Being,' eng, *Cyberpsychology, Behavior and Social Networking*, vol. 18, no. 7, pp. 411–416, Jul. 2015, ISSN: 2152-2723. DOI: 10.1089/cyber.2014.0487.

[4] S. Kang, W. Choi, C. Y. Park *et al.*, 'K-EmoPhone: A Mobile and Wearable Dataset with In-Situ Emotion, Stress, and Attention Labels,' en, *Scientific Data*, vol. 10, no. 1, p. 351, Jun. 2023, Publisher: Nature Publishing Group, ISSN: 2052-4463. DOI: 10.1038/s41597-023-02248-2. [Online]. Available: https://www.nature.com/articles/s41597-023-02248-2 (visited on 13/05/2024).

[5] J. Park and K. Cho, 'Smartphone usage patterns and their impact on mental health: Insights from the k-emophone dataset,' *Korean Journal of Psychology*, vol. 14, no. 1, pp. 77–92, 2023.

[6] R. Wei, 'Motivations for using the mobile phone for mass communications and entertainment,' *Telematics and Informatics*, vol. 25, no. 1, pp. 36–46, Feb. 2008, ISSN: 0736-5853. DOI: 10.1016/j.tele.2006.03.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736585306000177 (visited on 13/05/2024).

[7] S. Melumad and M. T. Pham, 'The Smartphone as a Pacifying Technology,' *Journal of Consumer Research*, vol. 47, no. 2, pp. 237–255, Aug. 2020, ISSN: 0093-5301. DOI: 10.1093/jcr/ucaa005. [Online]. Available: https://doi.org/10.1093/jcr/ucaa005 (visited on 13/05/2024).

[8] A. Lepp, 'The Intersection of Cell Phone Use and Leisure: A Call for Research,' *Journal of Leisure Research*, vol. 46, no. 2, pp. 218–225, Apr. 2014, Publisher: Routledge, ISSN: 0022-2216. DOI: 10.1080/00222216.2014.11950321. [Online]. Available: https://doi.org/10.1080/00222216.2014.11950321 (visited on 13/05/2024).

[9] S. Kaya and M. Argan, 'An exploratory qualitative study on mobile leisure (m-leisure): A case of mobile phone users in Turkey,' en, *İnternet Uygulamaları ve Yönetimi Dergisi*, vol. 6, no. 1, pp. 21–32, May 2015, Number: 1 Publisher: Pamukkale Üniversitesi, ISSN: 1309-8810. DOI: 10.5505/iuyd.2015.47955. [Online]. Available: https://dergipark.org.tr/tr/pub/iuyd/issue/34115/377377 (visited on 13/05/2024).

[10] A. Lepp, J. Li, J. E. Barkley and S. Salehi-Esfahani, 'Exploring the relationships between college students' cell phone use, personality and leisure,' *Computers in Human Behavior*, vol. 43, pp. 210–219, Feb. 2015, ISSN: 0747-5632. DOI: 10.1016/j.chb.2014.11.006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0747563214005822 (visited on 13/05/2024).

[11] P. D.Allison, *Fixed Effects Regression Models*, en. SAGE Publications, Inc., 2009, ISBN: 978-1-4129-9386-9. DOI: 10.4135/9781412993869. [Online]. Available: https://methods.sagepub.com/book/fixed-effects-regression-models (visited on 13/06/2024).

6. Appendix

A.

```python
# necessary imports
import pandas as pd
import numpy as np
import os
import seaborn as sns
import matplotlib.pyplot as plt
```

Data Manipulation and Extraction

Table and column selection

Tables and respective fields needed:

- Smartphone Data:
  - AppUsageEvent, containing all the necessary information of apps used
    - name
    - category
  - Location, containing the coordinates to be able to determine where apps were used
    - longitude
    - latitude
- Wearable Smartwatch Data:
  - AmbientLight, the light intensity in lumen per square meter
    - brightness
- Self reported emotional and cognitive states Data
  - EsmResponse
    - stress
    - valence
    - arousal
    - attention
    - pcode (participant number)

```python
data_dir = os.path.join("..","data")

# Smartphone and Smartwatch data
table_names = ["AppUsageEvent", "Location", "AmbientLight",
"UltraViolet"]
data_files = [f"{name}.csv" for name in table_names]
column_names = ["app_name", "app_category", "brightness",
"uv_intensity", "longitude", "latitude"]
# ESM response data
esm_path = os.path.join(data_dir,"SubjData","EsmResponse.csv")
```

Functions used

To TimeSeries

This step is **crucial** to our preprocessing as it allows us to merge the different tables used to one another by using the merge_asof pandas method (which matches on nearest timestamp). It is also important for determining big gaps in the data (see *Getting home coordinates* Section).

- Converting timestamps to pandas.datetime objects
- Using datetime objects as dataframe index for TimeSeries purposes
- Converting index to Korean timezone

```
def df_to_timeseries(df, timestamp_col="timestamp"):
    df[timestamp_col] = pd.to_datetime(df[timestamp_col], unit="ms") # convert timestamp to datetime
    df.set_index(timestamp_col,inplace=True) # set it as index
    df = df.tz_localize("UTC") # need to localize a timezone
    df = df.tz_convert("Asia/Seoul") # convert it to Korean timezone
    return df
```

Feature Extraction *at_home*

These methods helped us determine whether app-usage-events occurred at participants' home or not. During our data exploration when looking at participants' heart rate (HR), measured by the wearable smartwatch, we noticed there were around 7 big gaps in the HR data for each participant. Knowing that participants had to charge their wearable devices every night we deduced that getting the coordinates of the last and first data recordings before and after each data gap would give us the participants' home coordinates (we used the ambiental light table from the wearabale data, as we were already planning to use it).

- Measures the time difference between each consecutive data recording
- Identifies the ones with a time difference that exceeds 4 hours (assuming participants charged their wearable devices at night, when at home)
- Finds the closest corresponding indices in the coordinates table and extracts the coordinates of each identified "gap"
- Returns average (they may differ a bit, depending on the room where they took the wearable off) of the extracted coordinates
- Used haversine distance function to determine how far each app-usage-event was performed from the determined home coordiantes

```
def get_home_coor(amb_df,loc_df):
    time_diff = amb_df.index.to_series().diff()
    # Find indices where the time difference exceeds 4 hours
    jump_indices = time_diff[time_diff > pd.Timedelta(hours=4)].index
    closest_indices = {}
    for jump_index in jump_indices:
```

```
        indexer = loc_df.index.get_indexer([jump_index])
        closest_index = loc_df.index[indexer[0]]
        closest_indices[jump_index] = closest_index

    # Extract values from other_df at the closest indices and get
their average
    return loc_df.loc[closest_indices.values()].mean()

def haversine(lon1, lat1, lon2, lat2): # Returns haversine distance
between two coordinates
    lon1, lat1, lon2, lat2 = map(np.radians, [lon1, lat1, lon2, lat2])
    dlon = lon2 - lon1
    dlat = lat2 - lat1
    a = np.sin(dlat/2)**2 + np.cos(lat1) * np.cos(lat2) *
np.sin(dlon/2)**2
    c = 2 * np.arcsin(np.sqrt(a))
    km = 6371 * c
    return km * 1000
```

Determining Entertainment app categories

The AppUsageEvent table contains a column called "category" which states the category of the application used. The values of this field were retrieved from Google Play and from application archive websites (i.e., https://apkcombo.com) for those that weren't available on Google Play, the rest were manually labeled (Kang et al., 2023).

Categories left out:

- PERSONALIZATION, COMMUNICATION, PHOTOGRAPHY, SYSTEM, FINANCE, TOOLS, PRODUCTIVITY, HEALTH_AND_FITNESS, TRAVEL_AND_LOCAL, MAPS_AND_NAVIGATION, LIFESTYLE, HOUSE_AND_HOME, ART_AND_DESIGN, FOOD_AND_DRINK, EDUCATION, BUSINESS, BEAUTY, AUTO_AND_VEHICLES, WEATHER
- *LIBRARIES_AND_DEMO* (After some data exploration, we decided to remove this category from our chosen ones as it contained only one app which was the participants' university portal)

Categories chosen as Entertainment related:

- **VIDEO_PLAYERS**,**MUSIC_AND_AUDIO**,**SOCIAL**, **GAME**, **SHOPPING**, **NEWS_AND_MAGAZINES**, **SPORTS**, **BOOKS_AND_REFERENCE**,**COMICS**
- ***ENTERTAINMENT*** (this category was too vague so we looked up what type of apps were included here and found apps that would correctly belong to other categories chosen, for example "Netflix" and other video players were moved to the VIDEO_PLAYER category, the same happened with some SHOPPING apps, BOOKS_AND_REFERENCE and SOCIAL)
- ***MISC*** (some apps from here were moved to COMICS and GAMES)

Manually defined category

- **GAME_RELATED** (many apps within the ENTERTAINMENT category were game related but were not games directly, this included game exchange platforms, game tools, game statistics, etc)

Create chosen categories list and mapping dictionary (collapsable)

```
entertainment_categories = ['SOCIAL','SHOPPING','BOOKS_AND_REFERENCE',
                            'COMICS','MUSIC_AND_AUDIO',
'GAME','VIDEO_PLAYERS',

'SPORTS','NEWS_AND_MAGAZINES','GAME_RELATED']

app_name_to_new_category = {
    'Netflix': 'VIDEO_PLAYER',
    'AfreecaTV': 'VIDEO_PLAYER',
    '   ': 'BOOKS_AND_REFERENCE',
    'Google Play  ': 'GAME_RELATED',
    '  ': 'SHOPPING',
    '  TV': 'VIDEO_PLAYER',
    'Twitch': 'VIDEO_PLAYER',
    '  ': 'GAME',
    'CGV': 'SHOPPING',
    '   ': 'BOOKS_AND_REFERENCE',
    ' ': 'MUSIC_AND_AUDIO',
    'U+ tv': 'VIDEO_PLAYER',
    'TVING': 'VIDEO_PLAYER',
    'JAM Live': 'VIDEO_PLAYER',
    'CashLeaflet': 'SHOPPING',
    '  ': 'SHOPPING',
    'Galaxy Apps': 'APP_STORE',
    'LoL  ': 'GAME_RELATED',
    'OP.GG': 'GAME_RELATED',
    'GGtics': 'GAME_RELATED',
    'CGV  ': 'SHOPPING',
    'Prime Video': 'VIDEO_PLAYER',
    ' ': 'VIDEO_PLAYER',
    '   ': 'SHOPPING',
    '   ': 'VIDEO_PLAYER',
    'Doctor Who': 'VIDEO_PLAYER',
    '  ': 'VIDEO_PLAYER',
    '    ': 'LIBRARIES_AND_DEMO',
    ' tv  ': 'VIDEO_PLAYER',
    '      ': 'VIDEO_PLAYER',
    'V LIVE': 'VIDEO_PLAYER',
    'FOW.KR': 'GAME_RELATED',
    'Steam': 'GAME_RELATED',
    'ㅁㅔ가ㅂㅏㄱㅅㅡ': 'SHOPPING',
    'Nintendo Switch Online': 'GAME_RELATED',
    '   ': 'SOCIAL_MEDIA',
    'HTV 3.4.6': 'VIDEO_PLAYER',
```

```
    'Q.Feat': 'VIDEO_PLAYER',
    'WoW BfA Talent Calculator': 'GAME_RELATED',
    '   4': 'GAME',
    '   2:   ': 'GAME',
    'Hentoid': 'COMICS',
}
```

Function to map apps to new categories

```
def map_entertainment_to_new_category(row):
    if row["category"] == "ENTERTAINMENT":
        return app_name_to_new_category.get(row["name"],
"ENTERTAINMENT")
    elif row["category"] == "MISC":
        return app_name_to_new_category.get(row["name"], "MISC")
    else:
        return row["category"]
```

Converting interval and continuous to categorical

All of our dependent variables (stress, valence, arousal and attention) were in an interval form, the values ranged from −3 to +3 (from not stressed at all to very stressed, for example). For easier interpretation of the data we turned this interval data into binary data (stressed or not stressed for example). Our hope was to turn the results from the statistical analysis to be more intuitive and actionable.

- **0** for values -3, -2, -1 and 0
- **1** for values 1, 2, 3

Brightness, the light intensity in lumen per square meter (lx), was originally a continuous variable, ranging from low to high levels of brightness. For easier interpretation of the data, we categorized brightness into distinct levels (e.g., low, medium, high).

- LOW: Less than 300 lx
- MEDIUM: Between 300 lx and 750 lx
- HIGH: Greater than 750 lx

```
def dv_to_binary(df, dv):
    df[dv] = df[dv].apply(lambda x: 0 if x <=0 else 1)

def brightness_to_categorical(df):
    def categorize_brightness(brightness):
        if brightness <= 300:
            return 'LOW'
        elif 300 < brightness < 750:
            return 'MEDIUM'
        else:
            return 'HIGH'
    df["brightness"] = df["brightness"].apply(categorize_brightness)
```

```
    return df
```

Manipulation and merging

```
P = {} # Dictionary that'll contain all participant data

# get in situ data and turn it to TimeSeries
dvs = ["stress", "valence","arousal","attention"]
esm_df =
df_to_timeseries(pd.read_csv(esm_path),timestamp_col="responseTime")
[dvs + ["pcode"]]
for dv in dvs:
    dv_to_binary(esm_df, dv)

for p_code in os.listdir(data_dir):
    if p_code.startswith("P"): # Only get Participant Directories
        pn = int(p_code[1:])

        # get tables (and columns) of interest and turn them to
TimeSeries

        # AppUsageEvent

app_df=df_to_timeseries(pd.read_csv(os.path.join(data_dir,p_code,"AppU
sageEvent.csv")))[["name","category"]]
        # apply the function to update the app_category column
        app_df["category"] =
app_df.apply(map_entertainment_to_new_category, axis=1)
        # filter for entertainment related apps only
        app_df =
app_df[app_df["category"].isin(entertainment_categories)]

        # AmbienLight
        amb_df =
df_to_timeseries(pd.read_csv(os.path.join(data_dir,p_code,
"AmbientLight.csv")))
        # apply brightness_to_categorical
        amb_df = brightness_to_categorical(amb_df)

        # Location
        loc_df =
df_to_timeseries(pd.read_csv(os.path.join(data_dir,p_code,
"Location.csv")))
        loc_df = loc_df[loc_df["accuracy"] < 20]
[["longitude","latitude"]] # remove innacurate readings

        # merge using merge_asof (we match on nearest timestamp rather
than equal timestamps)
        joined_df = pd.merge_asof(app_df, amb_df,
```

```
left_index=True,right_index=True)
        joined_df = pd.merge_asof(joined_df, loc_df,
left_index=True,right_index=True)
        #joined_df.columns = column_names

        # calculate distance from determined home
        home_longitude, home_latitude = get_home_coor(amb_df,loc_df)
        distances = haversine(joined_df['longitude'],
joined_df['latitude'], home_longitude, home_latitude)
        joined_df["at_home"] = distances <= 25


        final_df=pd.merge_asof(joined_df, esm_df[esm_df["pcode"] ==
p_code], left_index=True,right_index=True)
        P[pn] = final_df.drop(["longitude", "latitude"],axis=1)
```

Saving preprocessed data

```
# all data
df = pd.concat(P.values(), axis = 0)
df.to_csv(os.path.join("clean_data", "final_data.csv"))
```

Results

June 12, 2024

[1]:
```
# necessary imports
import pandas as pd
import numpy as np
import os
import seaborn as sns
import matplotlib.pyplot as plt
import statsmodels.formula.api as smf
from io import StringIO
```

0.0.1 Load Data

[16]:
```
df = pd.read_csv(os.path.join("clean_data", "final_data.csv"))
```

0.1 Descriptive Statistics

[19]:
```
# number of data points
df.shape[0]
```

[19]:
```
193068
```

[47]:
```
df.describe(include="all")
```

[47]:
```
                               timestamp      name category brightness  \
count                             193068    193068   193068     191135
unique                            193049       268       10          3
top     2019-05-16 11:09:33.834000+09:00  Facebook   SOCIAL        LOW
freq                                   2     47637   109561     178294
mean                                 NaN       NaN      NaN        NaN
std                                  NaN       NaN      NaN        NaN
min                                  NaN       NaN      NaN        NaN
25%                                  NaN       NaN      NaN        NaN
50%                                  NaN       NaN      NaN        NaN
75%                                  NaN       NaN      NaN        NaN
max                                  NaN       NaN      NaN        NaN

       at_home         stress        valence        arousal      attention  \
count   193068  190453.000000  190453.000000  190453.000000  190453.000000
unique       2            NaN            NaN            NaN            NaN
```

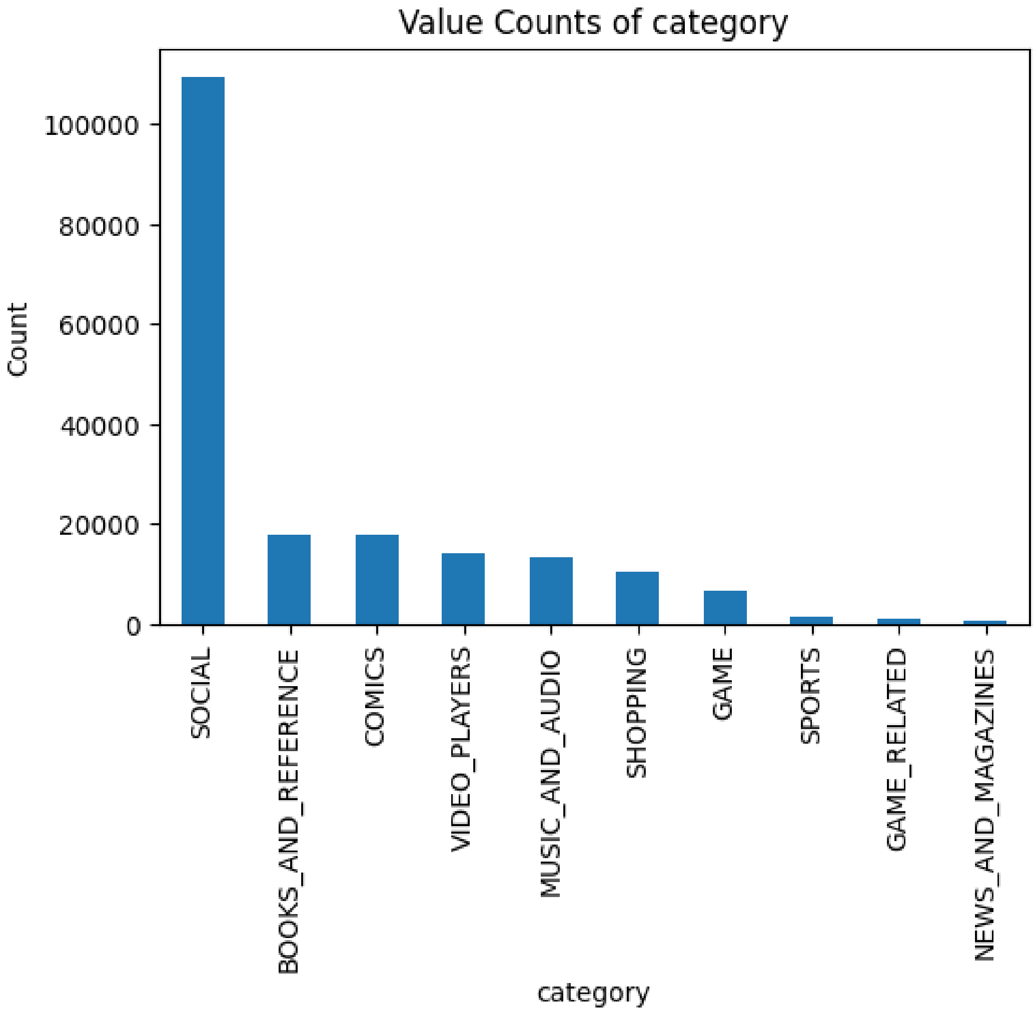
```
top       False         NaN         NaN         NaN         NaN
freq     154641         NaN         NaN         NaN         NaN
mean        NaN    0.312823    0.550141    0.366663    0.472095
std         NaN    0.463644    0.497481    0.481895    0.499222
min         NaN    0.000000    0.000000    0.000000    0.000000
25%         NaN    0.000000    0.000000    0.000000    0.000000
50%         NaN    0.000000    1.000000    0.000000    0.000000
75%         NaN    1.000000    1.000000    1.000000    1.000000
max         NaN    1.000000    1.000000    1.000000    1.000000

          pcode
count    190453
unique       77
top         P56
freq      11036
mean        NaN
std         NaN
min         NaN
25%         NaN
50%         NaN
75%         NaN
max         NaN
```

0.1.1 Distributions

[37]:
```
def plot_distribution_of_values(col):
    plt.figure(figsize=(6, 4))
    df[col].value_counts().plot(kind='bar')
    plt.title(f'Value Counts of {col}')
    plt.xlabel(col)
    plt.ylabel('Count')
    plt.show();
```

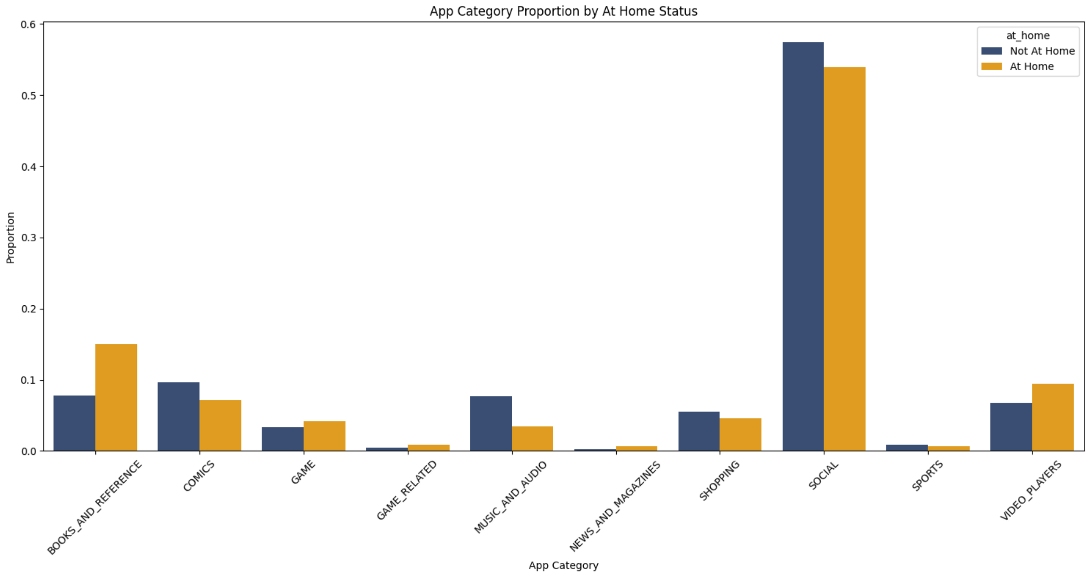
category

[38]:
```
plot_distribution_of_values("category")
```



[46]:
```
# Calculate the distributions
at_home_dist = df[df["at_home"]]["category"].value_counts() / df[df["at_home"]].
  ↪shape[0]
not_at_home_dist = df[~df["at_home"]]["category"].value_counts() /␣
  ↪df[~df["at_home"]].shape[0]

# Create a DataFrame for plotting
dist_df = pd.DataFrame({
    'category': at_home_dist.index.tolist() + not_at_home_dist.index.tolist(),
    'proportion': at_home_dist.tolist() + not_at_home_dist.tolist(),
    'at_home': ['At Home'] * len(at_home_dist) + ['Not At Home'] *␣
  ↪len(not_at_home_dist)
}).sort_values(by=["proportion"], ascending=False)
```

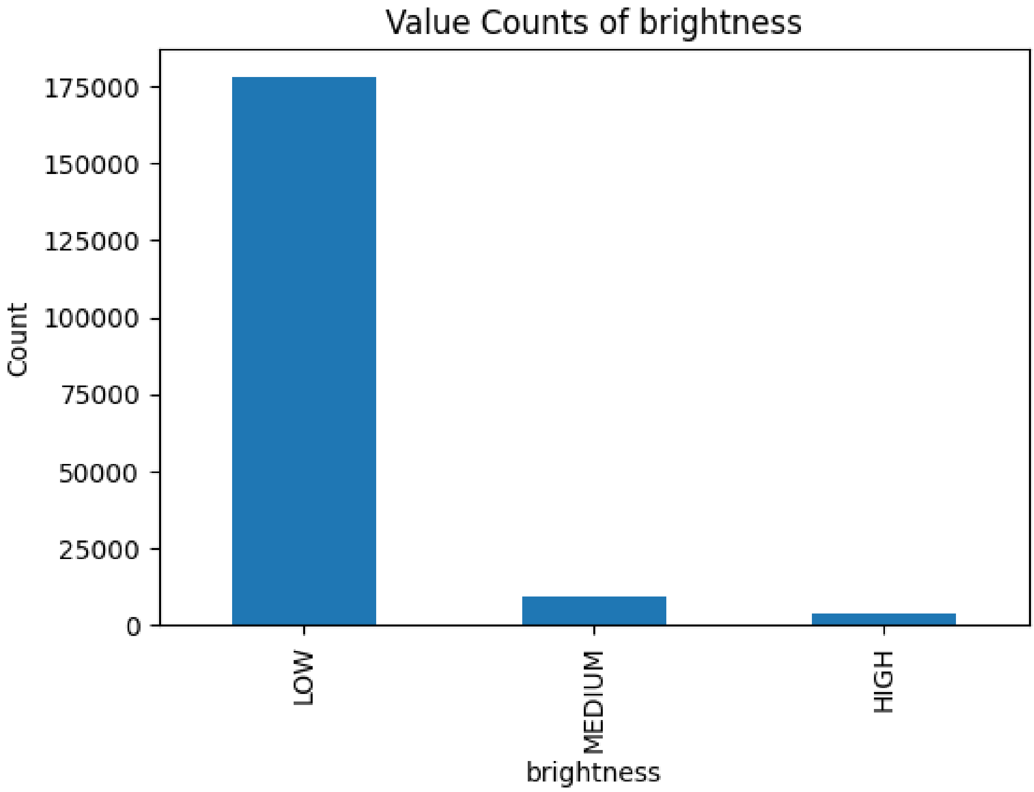
```
# Sort the DataFrame by app_category
dist_df = dist_df.sort_values(by=['category', 'at_home'], ascending=[True,␣
↪False])

# Plot the data
plt.figure(figsize=(15, 8))
sns.barplot(x='category', y='proportion', hue='at_home', data=dist_df,␣
↪palette=['#2f4b7c', '#ffa600'])
plt.xlabel('App Category')
plt.ylabel('Proportion')
plt.title('App Category Proportion by At Home Status')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



brightness and at_home

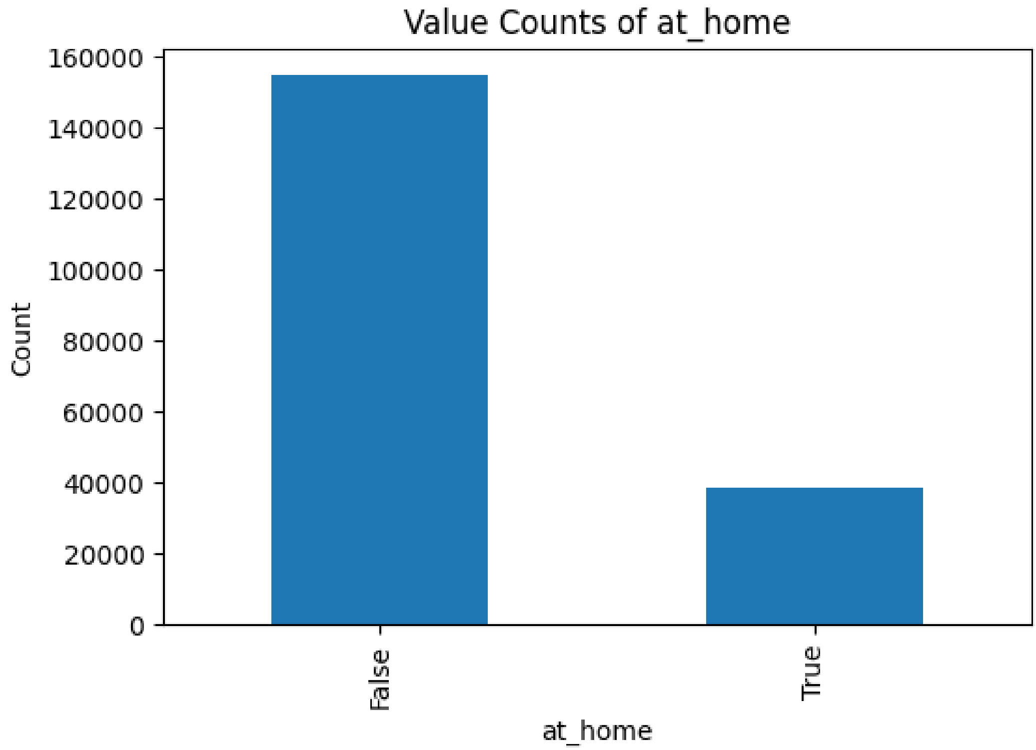
[39]:
```
plot_distribution_of_values("brightness")
```

L. 383



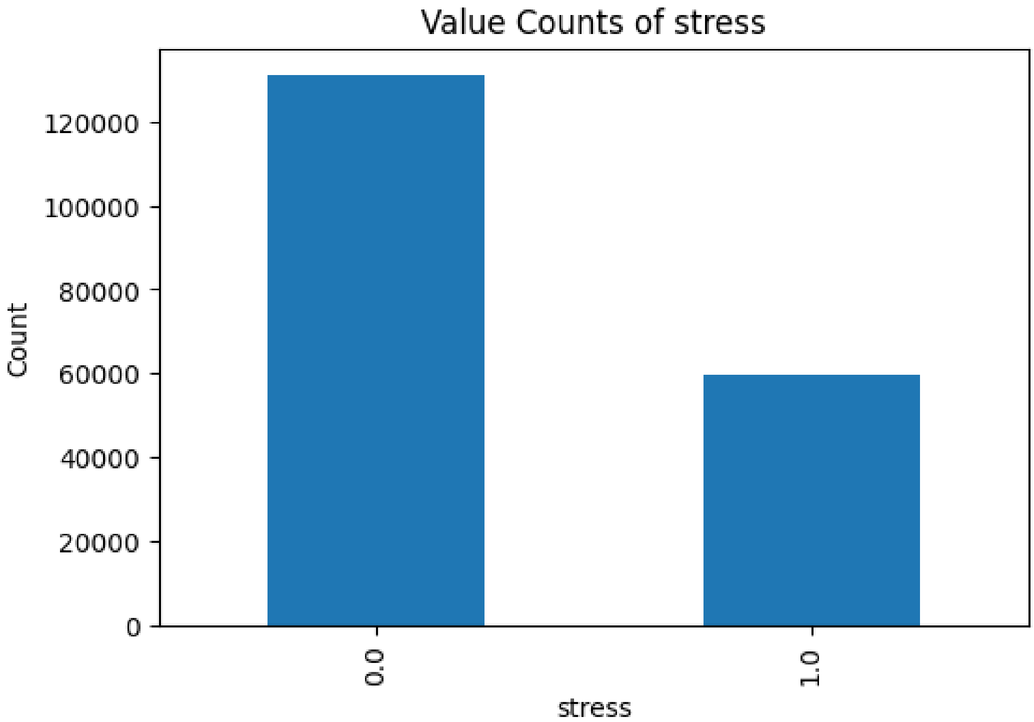
[40]:
```
plot_distribution_of_values("at_home")
```

5



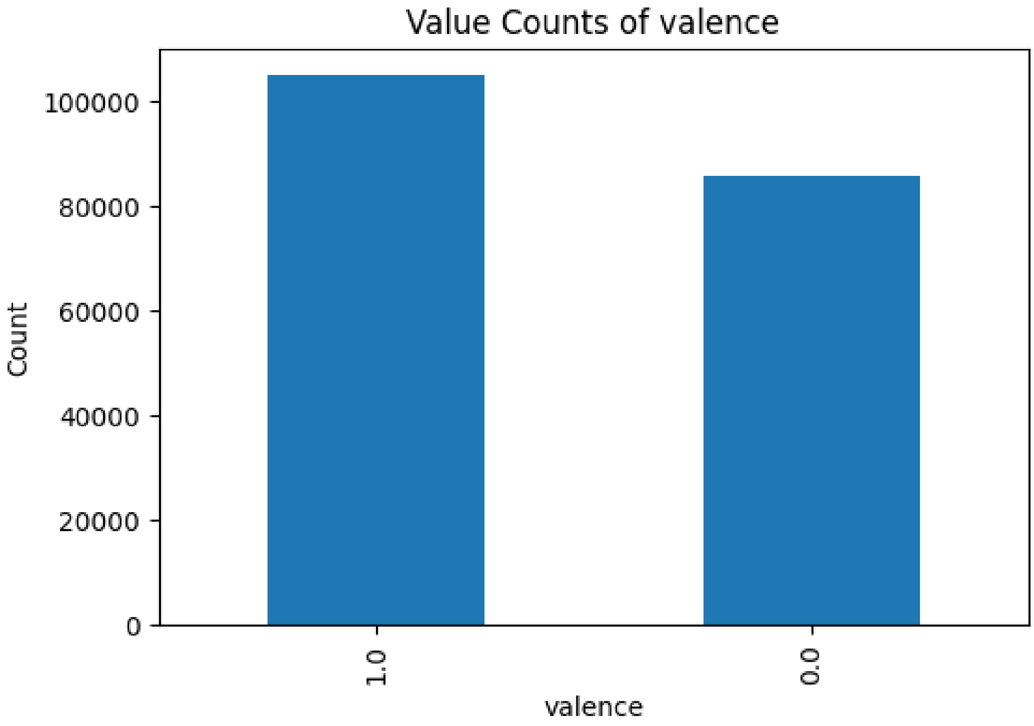
stress, valence, arousal and attention

[41]:
```
plot_distribution_of_values("stress")
```



[42]:
```
plot_distribution_of_values("valence")
```

0. 386

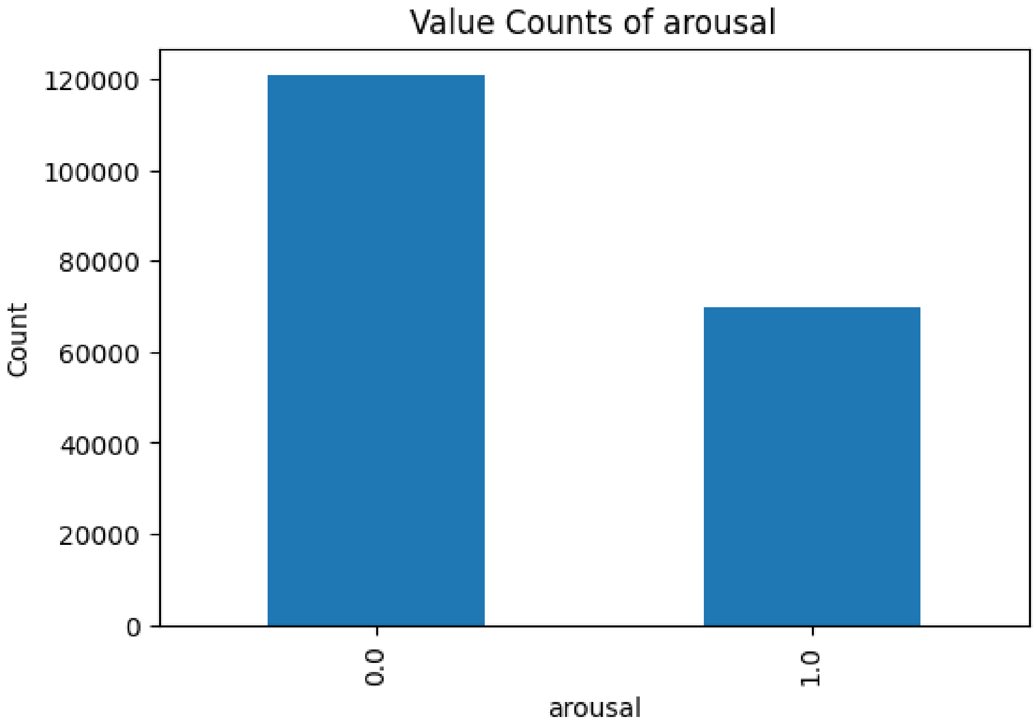


[43]:

```
plot_distribution_of_values("arousal")
```

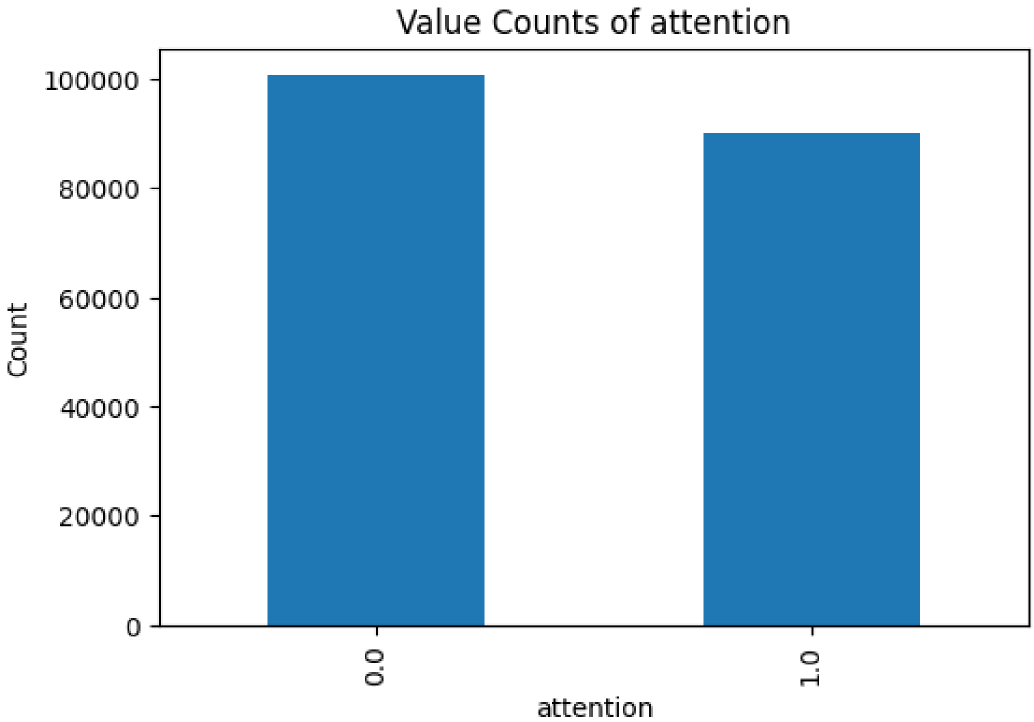
8

P. 387




```
[44]: plot_distribution_of_values("attention")
```

Q. 388



[9]:
```
df.describe(include="all")
```

[9]:
```
                               timestamp      name category brightness  \
count                             191984    191984   191984     190051
unique                            191965       260        9          3
top     2019-05-21 22:53:44.760000+09:00  Facebook   SOCIAL        LOW
freq                                   2     47637   109561     177275
mean                                 NaN       NaN      NaN        NaN
std                                  NaN       NaN      NaN        NaN
min                                  NaN       NaN      NaN        NaN
25%                                  NaN       NaN      NaN        NaN
50%                                  NaN       NaN      NaN        NaN
75%                                  NaN       NaN      NaN        NaN
max                                  NaN       NaN      NaN        NaN

       at_home         stress        valence       arousal      attention  \
count   191984  189369.000000  189369.000000  189369.00000  189369.000000
unique       2            NaN            NaN           NaN            NaN
top      False            NaN            NaN           NaN            NaN
freq    153905            NaN            NaN           NaN            NaN
mean       NaN       0.312765       0.549520       0.36788       0.472231
std        NaN       0.463621       0.497543       0.48223       0.499230
```

```
min          NaN      0.000000      0.000000      0.00000      0.000000
25%          NaN      0.000000      0.000000      0.00000      0.000000
50%          NaN      0.000000      1.000000      0.00000      0.000000
75%          NaN      1.000000      1.000000      1.00000      1.000000
max          NaN      1.000000      1.000000      1.00000      1.000000

         pcode
count   189369
unique      77
top        P56
freq     11030
mean       NaN
std        NaN
min        NaN
25%        NaN
50%        NaN
75%        NaN
max        NaN
```

[ ]:

0.2 Logisitc Regressions with Fixed Effects

- Main Effects:
  - The models evaluates how app category, being at home and brightness levels influence whether people are stressed, feeling good (valence), attentive and aroused.
- Interaction Effects:
  - They also evaluate whether the impact of using certain app categories on stress, valence, arousal, attention is different when the participant is at home versus elsewhere, or based on brightness levels.
- Individual Differences:
  - The model controls for unique characteristics of each participant to get more accurate results.

Logistic regression is chosen because it is well-suited for modeling binary outcomes, such as whether a person is stressed (yes/no), feeling good (yes/no), attentive (yes/no), or aroused (yes/no). By using logistic regression, we can estimate the probability of these outcomes based on the predictors (app category, location, brightness) and their interactions. This method allows us to understand the relationship between the predictors and the binary outcomes while accounting for individual differences through fixed effects.

[89]:
```
def generate_model(dv):
    formula = f"{dv} ~ C(category) * at_home + C(pcode)"
    return smf.logit(formula=formula, data=df).fit()

def get_model_results(model):
    html_string = model.summary().tables[1].as_html() # Extract the HTML from␣
 ↪the model summary
```

```
    html_io = StringIO(html_string) # Use StringIO to wrap the HTML string
    df = pd.read_html(html_io, header=0)[0] # Read the HTML into a DataFrame
    df.columns = ["predictor"] + df.columns[1:].to_list() # Rename column
    return df[~df["predictor"].str.contains("pcode")]

def plot_forest(results_df):
    fig, ax = plt.subplots(figsize=(10, 8))
    ax.errorbar(results_df["coef"], results_df["predictor"], xerr=1.
 ↪96*results_df["std err"], fmt='o', color='black', ecolor='lightgray',␣
 ↪elinewidth=3, capsize=0)
    # Add vertical line at zero
    ax.axvline(x=0, linestyle='--', color='red')
    # Add labels and title
    ax.set_xlabel('Coefficient')
    ax.set_ylabel('Predictor')
    ax.set_title('Forest Plot of Coefficient Estimates')
    # Show plot
    plt.show()
```

[ ]:
```
df = stress_df[stress_df["P>|z|"] < .05]
# Plot
fig, ax = plt.subplots(figsize=(10, 8))

ax.errorbar(df["coef"], df["Unnamed: 0"], xerr=1.96*df["std err"], fmt='o',␣
 ↪color='black', ecolor='lightgray', elinewidth=3, capsize=0)

# Add vertical line at zero
ax.axvline(x=0, linestyle='--', color='red')

# Add labels and title
ax.set_xlabel('Coefficient')
ax.set_ylabel('Variable')
ax.set_title('Forest Plot of Coefficient Estimates')

# Show plot
plt.show()
```

0.2.1 Stress

[76]:
```
stress_model = generate_model("stress")
stress_results = get_model_results(stress_model)
```

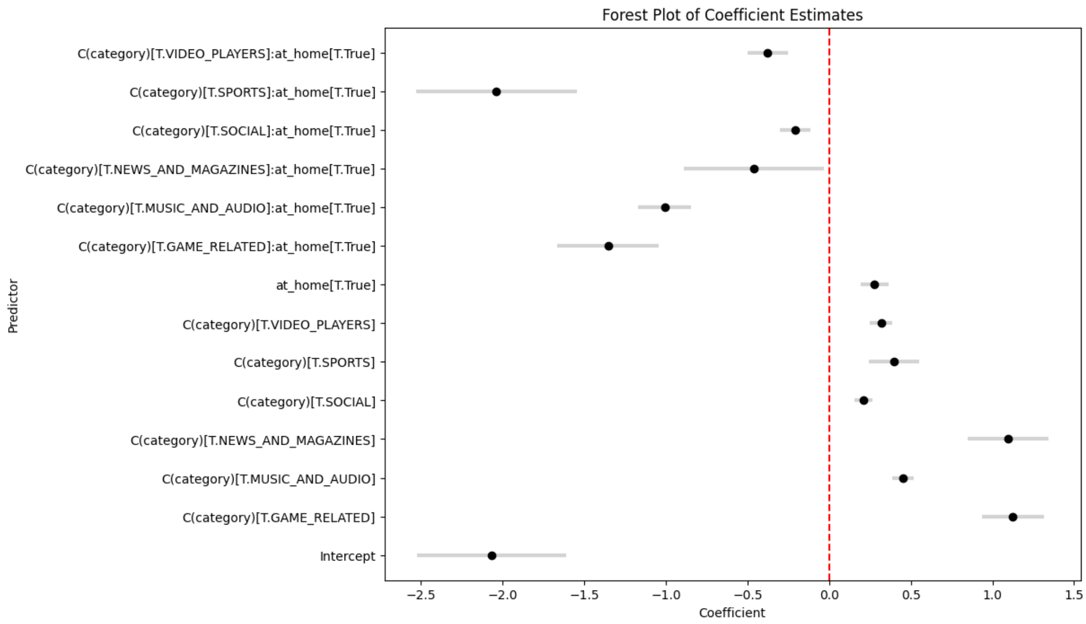
```
Optimization terminated successfully.
         Current function value: 0.499745
         Iterations 7
```

T. 391

[77]:
```
# select only statistacally significant effects
stress_results[stress_results["P>|z|"] < .05]
```

[77]:
```
                                       predictor    coef  std err  \
0                                      Intercept -2.0686    0.233
3                   C(category)[T.GAME_RELATED]  1.1255    0.096
4                 C(category)[T.MUSIC_AND_AUDIO]  0.4531    0.034
5              C(category)[T.NEWS_AND_MAGAZINES]  1.0969    0.127
7                         C(category)[T.SOCIAL]  0.2101    0.027
8                         C(category)[T.SPORTS]  0.3975    0.078
9                  C(category)[T.VIDEO_PLAYERS]  0.3184    0.036
10                              at_home[T.True]  0.2793    0.043
89     C(category)[T.GAME_RELATED]:at_home[T.True] -1.3528    0.158
90   C(category)[T.MUSIC_AND_AUDIO]:at_home[T.True] -1.0072    0.083
91 C(category)[T.NEWS_AND_MAGAZINES]:at_home[T.True] -0.4586    0.219
93           C(category)[T.SOCIAL]:at_home[T.True] -0.2085    0.048
94           C(category)[T.SPORTS]:at_home[T.True] -2.0368    0.251
95    C(category)[T.VIDEO_PLAYERS]:at_home[T.True] -0.3770    0.063

         z  P>|z|  [0.025  0.975]
0   -8.863  0.000  -2.526  -1.611
3   11.721  0.000   0.937   1.314
4   13.521  0.000   0.387   0.519
5    8.611  0.000   0.847   1.347
7    7.642  0.000   0.156   0.264
8    5.121  0.000   0.245   0.550
9    8.768  0.000   0.247   0.390
10   6.528  0.000   0.195   0.363
89  -8.542  0.000  -1.663  -1.042
90 -12.128  0.000  -1.170  -0.844
91  -2.092  0.036  -0.888  -0.029
93  -4.388  0.000  -0.302  -0.115
94  -8.114  0.000  -2.529  -1.545
95  -6.003  0.000  -0.500  -0.254
```

[90]:
```
plot_forest(stress_results[stress_results["P>|z|"] < .05])
```



0.2.2 Valence

[78]:
```
valence_model = generate_model("valence")
valence_results = get_model_results(valence_model)
```

```
Warning: Maximum number of iterations has been exceeded.
         Current function value: 0.534613
         Iterations: 35
```

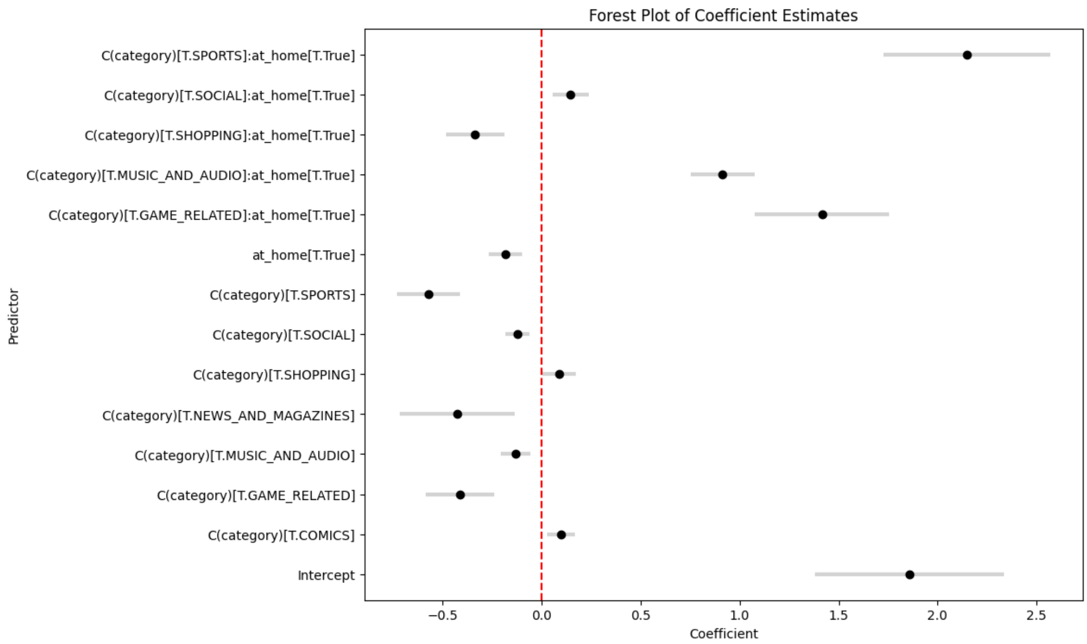
```
/Users/mauro/.pyenv/versions/3.10.6/envs/sandbox/lib/python3.10/site-
packages/statsmodels/base/model.py:604: ConvergenceWarning: Maximum Likelihood
optimization failed to converge. Check mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "
```

[79]:
```
# select only statistacally significant effects
valence_results[valence_results["P>|z|"] < .05]
```

[79]:
```
                             predictor    coef  std err      z  \
0                            Intercept  1.8564    0.243  7.637
1               C(category)[T.COMICS]  0.0979    0.036  2.729
3         C(category)[T.GAME_RELATED] -0.4108    0.088 -4.649
4      C(category)[T.MUSIC_AND_AUDIO] -0.1291    0.038 -3.429
5   C(category)[T.NEWS_AND_MAGAZINES] -0.4251    0.148 -2.864
6             C(category)[T.SHOPPING]  0.0897    0.043  2.085
7               C(category)[T.SOCIAL] -0.1234    0.031 -4.004
8               C(category)[T.SPORTS] -0.5696    0.081 -7.029
10                    at_home[T.True] -0.1817    0.043 -4.230
```

```
89     C(category)[T.GAME_RELATED]:at_home[T.True]  1.4159    0.173   8.173
90  C(category)[T.MUSIC_AND_AUDIO]:at_home[T.True]  0.9147    0.082  11.153
92         C(category)[T.SHOPPING]:at_home[T.True] -0.3355    0.075  -4.446
93           C(category)[T.SOCIAL]:at_home[T.True]  0.1467    0.046   3.219
94           C(category)[T.SPORTS]:at_home[T.True]  2.1464    0.215   9.981

    P>|z|  [0.025  0.975]
0   0.000   1.380   2.333
1   0.006   0.028   0.168
3   0.000  -0.584  -0.238
4   0.001  -0.203  -0.055
5   0.004  -0.716  -0.134
6   0.037   0.005   0.174
7   0.000  -0.184  -0.063
8   0.000  -0.728  -0.411
10  0.000  -0.266  -0.098
89  0.000   1.076   1.755
90  0.000   0.754   1.075
92  0.000  -0.483  -0.188
93  0.001   0.057   0.236
94  0.000   1.725   2.568
```

[91]:
```
plot_forest(valence_results[valence_results["P>|z|"] < .05])
```



0.2.3 Arousal

[80]:
```
arousal_model = generate_model("arousal")
arousal_results = get_model_results(arousal_model)
```

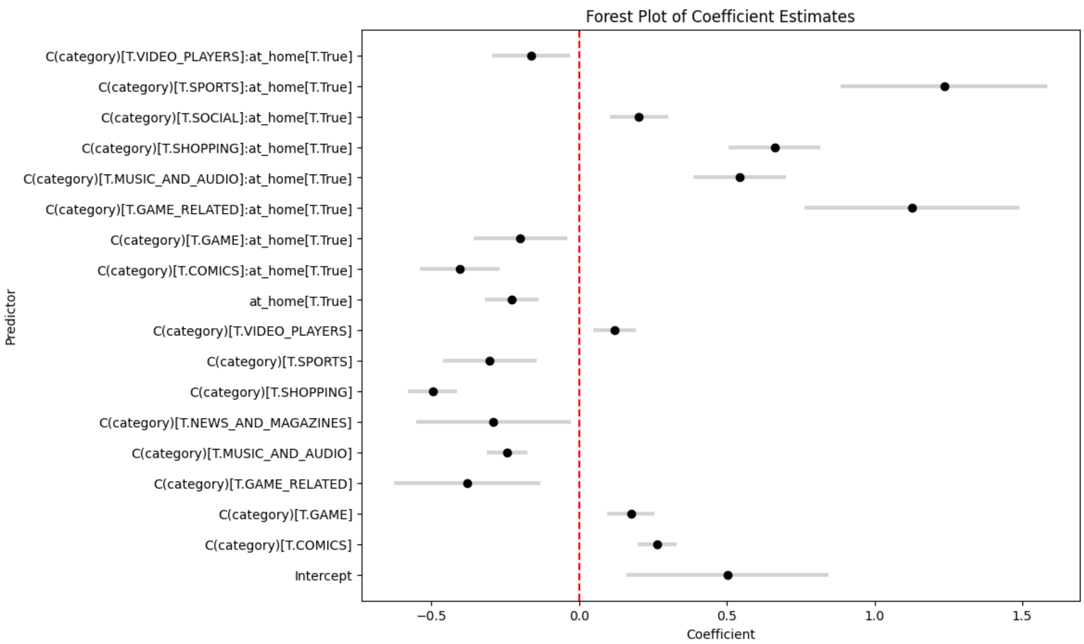
```
Optimization terminated successfully.
         Current function value: 0.525153
         Iterations 9
```

[81]:
```
# select only statistacally significant effects
arousal_results[arousal_results["P>|z|"] < .05]
```

[81]:
```
                                    predictor    coef  std err       z  \
0                                   Intercept  0.5020    0.174   2.885
1                     C(category)[T.COMICS]  0.2638    0.034   7.763
2                       C(category)[T.GAME]  0.1753    0.041   4.314
3               C(category)[T.GAME_RELATED] -0.3792    0.126  -3.016
4            C(category)[T.MUSIC_AND_AUDIO] -0.2440    0.035  -6.932
5         C(category)[T.NEWS_AND_MAGAZINES] -0.2897    0.134  -2.169
6                   C(category)[T.SHOPPING] -0.4967    0.042 -11.964
8                     C(category)[T.SPORTS] -0.3029    0.081  -3.752
9              C(category)[T.VIDEO_PLAYERS]  0.1193    0.037   3.248
10                             at_home[T.True] -0.2281    0.047  -4.861
87           C(category)[T.COMICS]:at_home[T.True] -0.4038    0.069  -5.848
88             C(category)[T.GAME]:at_home[T.True] -0.1995    0.081  -2.457
89     C(category)[T.GAME_RELATED]:at_home[T.True]  1.1271    0.185   6.109
90  C(category)[T.MUSIC_AND_AUDIO]:at_home[T.True]  0.5428    0.080   6.814
92         C(category)[T.SHOPPING]:at_home[T.True]  0.6619    0.079   8.379
93           C(category)[T.SOCIAL]:at_home[T.True]  0.2025    0.051   4.010
94           C(category)[T.SPORTS]:at_home[T.True]  1.2355    0.178   6.957
95    C(category)[T.VIDEO_PLAYERS]:at_home[T.True] -0.1615    0.067  -2.408

    P>|z|  [0.025  0.975]
0   0.004   0.161   0.843
1   0.000   0.197   0.330
2   0.000   0.096   0.255
3   0.003  -0.626  -0.133
4   0.000  -0.313  -0.175
5   0.030  -0.552  -0.028
6   0.000  -0.578  -0.415
8   0.000  -0.461  -0.145
9   0.001   0.047   0.191
10  0.000  -0.320  -0.136
87  0.000  -0.539  -0.268
88  0.014  -0.359  -0.040
89  0.000   0.766   1.489
90  0.000   0.387   0.699
92  0.000   0.507   0.817
```

```
93  0.000   0.104   0.301
94  0.000   0.887   1.584
95  0.016  -0.293  -0.030
```

[92]:
```python
plot_forest(arousal_results[arousal_results["P>|z|"] < .05])
```



0.2.4 Attention

[82]:
```python
attention_model = generate_model("attention")
attention_results = get_model_results(attention_model)
```

```
Optimization terminated successfully.
         Current function value: 0.582969
         Iterations 8
```

[84]:
```python
# select only statistacally significant effects
attention_results[attention_results["P>|z|"] < .05]
```

[84]:

```
                             predictor    coef  std err  \
0                            Intercept  1.5156    0.218
1                  C(category)[T.COMICS]  0.2630    0.033
2                    C(category)[T.GAME] -0.1851    0.040
4        C(category)[T.MUSIC_AND_AUDIO] -0.1097    0.034
5     C(category)[T.NEWS_AND_MAGAZINES] -0.4351    0.127
6                C(category)[T.SHOPPING] -0.1021    0.039
8                  C(category)[T.SPORTS] -0.3133    0.071
```

17

```
87               C(category)[T.COMICS]:at_home[T.True]  -0.5987    0.063
89         C(category)[T.GAME_RELATED]:at_home[T.True]   0.4449    0.139
90      C(category)[T.MUSIC_AND_AUDIO]:at_home[T.True]   0.9929    0.076
91  C(category)[T.NEWS_AND_MAGAZINES]:at_home[T.True]   1.2238    0.232
92             C(category)[T.SHOPPING]:at_home[T.True]   0.1981    0.071
93               C(category)[T.SOCIAL]:at_home[T.True]   0.1598    0.041
94               C(category)[T.SPORTS]:at_home[T.True]   3.0593    0.392

          z  P>|z|  [0.025  0.975]
0     6.941  0.000   1.088   1.944
1     8.044  0.000   0.199   0.327
2    -4.630  0.000  -0.264  -0.107
4    -3.260  0.001  -0.176  -0.044
5    -3.415  0.001  -0.685  -0.185
6    -2.607  0.009  -0.179  -0.025
8    -4.434  0.000  -0.452  -0.175
87   -9.490  0.000  -0.722  -0.475
89    3.199  0.001   0.172   0.717
90   13.081  0.000   0.844   1.142
91    5.282  0.000   0.770   1.678
92    2.800  0.005   0.059   0.337
93    3.890  0.000   0.079   0.240
94    7.810  0.000   2.292   3.827
```

[93]:
```python
plot_forest(attention_results[attention_results["P>|z|"] < .05])
```