

ChatGPT tools for R programming

Leon Lombaard & Megan Carnell



Introduction

Generative AI technology behind ChatGPT

API Keys

Packages

askgpt

Rtutor

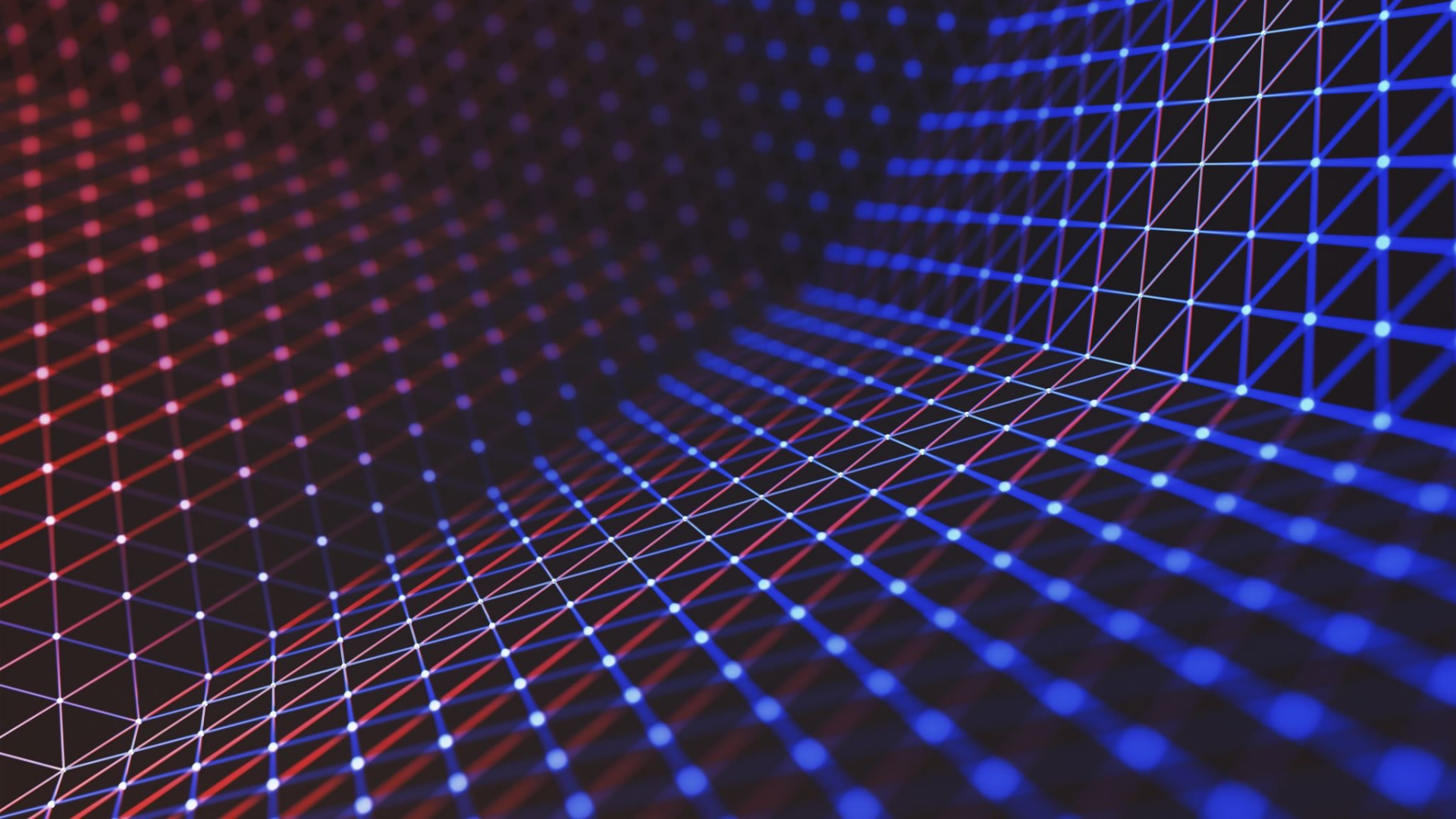
gptchatteR

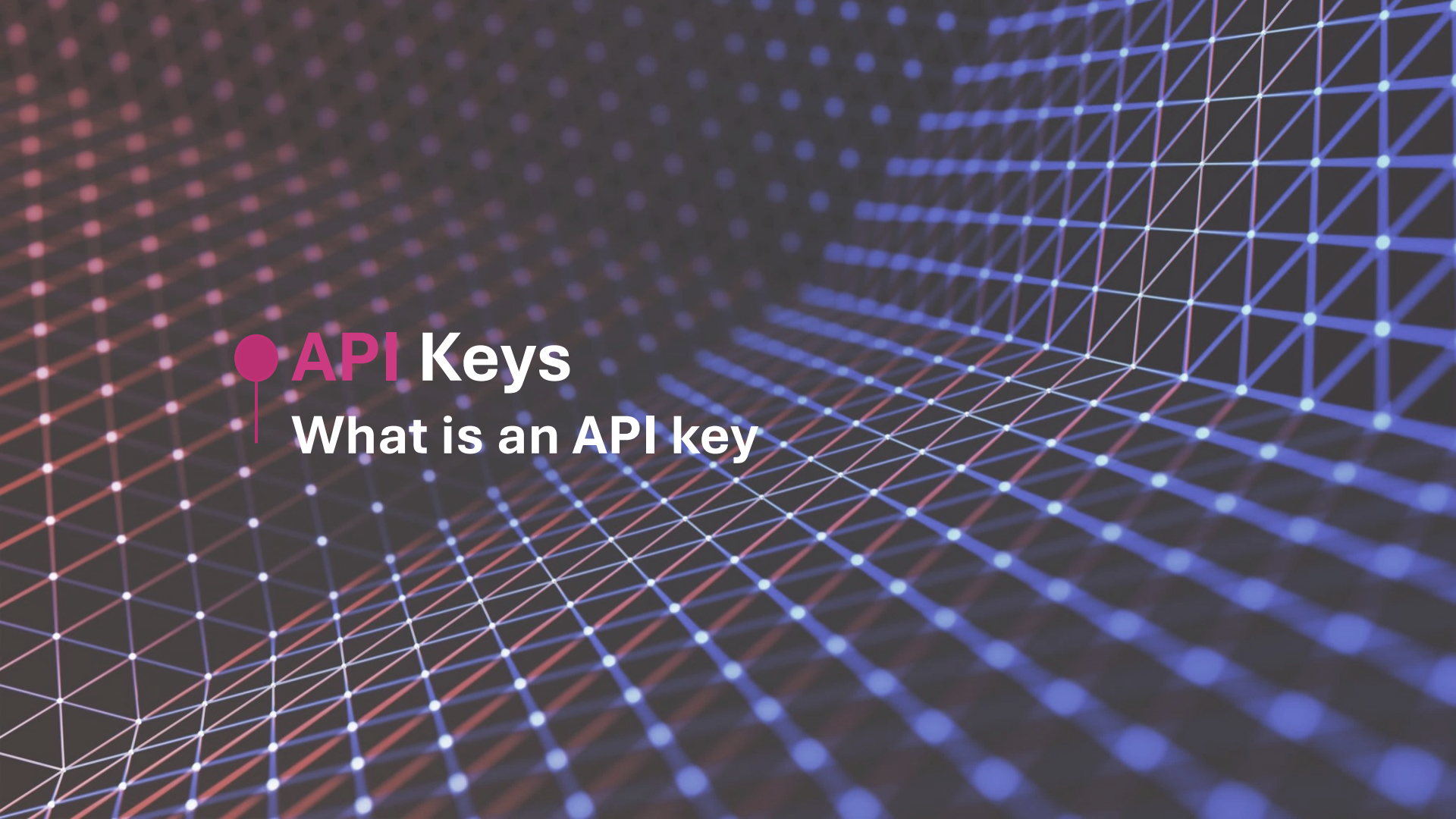
OpenAIR

Gemini

Caution!

Let's compare





● API Keys

What is an API key





● API Keys

How to get an API key

API Keys

How to get an API key



Create an
OpenAI account

Go to your profile > User
API keys > create your own

Login using
`login("YOURKEY")`

+ Create new secret key



● Packages

askgpt

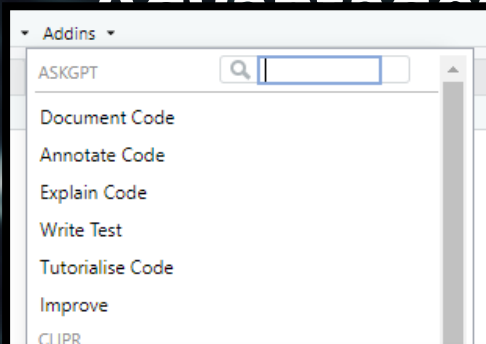
i It looks like you have not provided an API key yet.

1. Go to <https://askgpt.ai>
2. Log into your account
3. On the site, click the button + Create new secret key to create an API key
4. Copy this key into R/RStudio

```
askgpt("How do I make a bar chart?")
```


askgpt

Advantages



Hi, this is askgpt ☺.

- To start error logging, run ``log_init()`` now.
- To see what you can do use ``?askgpt()``.
- Or just run ``askgpt()`` with any question you want!

Report test coverage for a file
Report test coverage for a package
Document a package
Run examples
GPTSTUDIO
Spelling and Grammar
Comment your code
Chat

`annotate_code()`

`test_function()`

Example

— Answer —

Sure! Here is an example R code to create a dummy dataframe containing heart rates of 50 patients over a period of 5 days:

```
```R
Set seed for reproducibility
set.seed(123)

Create dummy data
patient_ids <- 1:50
heart_rates <- matrix(sample(60:100, 50*5, replace = TRUE), ncol = 5)
colnames(heart_rates) <- c("Day1", "Day2", "Day3", "Day4", "Day5")

Create dataframe
heart_rate_data <- data.frame(Patient_ID = patient_ids, heart_rates)

Print dataframe
print(heart_rate_data)
```
```

This code will generate a dummy dataframe called `heart\_rate\_data` with 50 rows (one for each patient) and 6 columns (Patient_ID, Day1 to Day5) containing random heart rate values for each patient over a period of 5 days.

Feel free to customize the code further if needed or let me know any specific requirements you have!

```
> |
```


Example

| Patient ID | Day 1 | Day 2 | Day 3 | Day 4 | Day 5 |
|------------|-------------|-------------|-------------|-------------|-------------|
| Patient_1 | 70.51619483 | 77.02654811 | 69.31674749 | 81.30191078 | 92.59048279 |
| Patient_2 | 73.15858008 | 74.77162596 | 77.05506967 | 81.15233793 | 85.49930381 |
| Patient_3 | 87.46966651 | 74.65703634 | 73.02646497 | 77.65762063 | 72.87883955 |
| Patient_4 | 75.56406713 | 85.94881827 | 72.2196592 | 66.93298713 | 79.34555247 |

Rtutor

3 Steps

Go to
<https://rtutor.ai/>

Upload a
dataset

Ask a
question

The screenshot shows the Rtutor web interface. At the top right is a blue 'Reset' button. Below it, the interface is divided into two main sections: 'Data' and 'File Upload'. The 'Data' section contains a dropdown menu currently showing 'mpg (examples)'. The 'File Upload' section contains a 'Browse...' button and a 'No file sele' button. Below these sections is a large text input area with a placeholder text: 'Examples:'. At the bottom of the interface, there is a red 'Submit' button, a 'Settings' button, and a checkbox labeled 'Python'.

RTutor Home EDA Report About

Basic Summary Table1 Categorical Numerical Correlation GGpairs EDA Reports

Data structure: df

```
tibble [234 x 11] (S3: tbl_df/tbl/data.frame)
 $ maker      : chr [1:234] "audi" "audi" "audi" "audi" ...
 $ model      : chr [1:234] "a4" "a4" "a4" "a4" ...
 $ dis        : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
 $ year       : Factor w/ 2 levels "1999","2008": 1 1 2 2 1 1 2 1 1 2 ...
 $ cylinder    : Factor w/ 4 levels "4","5","6","8": 1 1 1 1 3 3 3 1 1 1 ...
 $ transmission: chr [1:234] "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
 $ drive       : Factor w/ 3 levels "4","f","r": 2 2 2 2 2 2 1 1 1 1 ...
 $ city        : int [1:234] 18 21 20 21 16 18 18 16 20 ...
 $ highway     : int [1:234] 29 29 31 30 26 26 27 26 25 28 ...
 $ fuel        : Factor w/ 5 levels "c","d","e","p",...: 4 4 4 4 4 4 4 4 4 4 ...
 $ type        : chr [1:234] "compact" "compact" "compact" "compact" ...
```

Data summary: df

| maker | model | dis | year | cylinder | transmission | drive | city | highway | fuel | type |
|------------------|------------------|---------------|----------|----------|------------------|-------|---------------|---------------|-------|------------------|
| Length:234 | Length:234 | Min. :1.600 | 1999:117 | 4:81 | Length:234 | 4:103 | Min. : 9.00 | Min. :12.00 | c: 1 | Length:234 |
| Class :character | Class :character | 1st Qu.:2.400 | 2008:117 | 5: 4 | Class :character | f:106 | 1st Qu.:14.00 | 1st Qu.:18.00 | d: 5 | Class :character |
| Mode :character | Mode :character | Median :3.300 | | 6:79 | Mode :character | r: 25 | Median :17.00 | Median :24.00 | e: 8 | Mode :character |
| | | Mean :3.472 | | 8:70 | | | Mean :16.86 | Mean :23.44 | p: 52 | |
| | | 3rd Qu.:4.600 | | | | | 3rd Qu.:19.00 | 3rd Qu.:27.00 | r:168 | |
| | | Max. :7.000 | | | | | Max. :35.00 | Max. :44.00 | | |

Dataset: mpg Reset

R: violin plot

Create a violin plot of city vs. type. Color by type.
Add data points with jitter.

Submit Settings ☐ Python

Q&A: Ask about the code, result, error, or statistic

Data Types Description

R1: 508 tokens, 3 second(s)
Total API Cost: \$0.019
GPT-4o, Temperature=0.2

☐ Comments & questions

AI generated code:

Chunk #1

```
library(ggplot2)

# Create the violin plot with jittered data points
ggplot(df, aes(x = city, y = type, fill = type)) +
  geom_violin(trim = FALSE) +
  geom_jitter(width = 0.2, size = 1, alpha = 0.6) +
  theme_minimal() +
  labs(title = "Violin Plot of City vs. Type",
       x = "City",
       y = "Type") +
  theme(legend.position = "none")
```

Results:

Violin Plot of City vs. Type

gptchatteR

4 Steps

```
remotes::install_github("isinaltinkaya/gptchatteR",  
  build_vignettes = TRUE, dependencies = TRUE)  
library(gptchatteR)
```

Install
package

API key
Authenticator

```
chatter.auth("YOUR KEY")
```

```
chatter.create(temperature = 0)
```

Launch
chat
session

Submit
argument

```
chatter.chat("Your input here")      /* Create a chat */  
chatter.feed()                       /* Stores initial query for use later */
```


OpenAIR

3 Steps

```
devtools::install_github("mutter/OpenAIR")  
library("TheOpenAIR")
```

```
openai_api_key("YOUR-API-KEY")
```

```
chat("Your input here")
```

ChatGPT vs AI

Remove all numeric characters from '5XC-2a09ujnap9o2q0MP'



Sure! Here is the string with all numeric characters removed:

`XC-aujnapoqMP`

ChatGPT

OpenAIR

```
> chat("Remove all numeric characters from '5XC-2a09ujnap9o2q0MP'")
To remove all numeric characters from the string '5XC-2a09ujnap9o2q0MP', you can use the gsub function in R. You can
substitute numeric characters with an empty string. Here's how you can achieve that:
> chat("what will the output be")
The output of removing all numeric characters from the string '5XC-2a09ujnap9o2q0MP' using the provided R code will be:

'XC-aujnapoqMP'
output_string <- gsub("[0-9]", "", input_string)
# Print the output string with numeric characters removed
print(output_string)
```

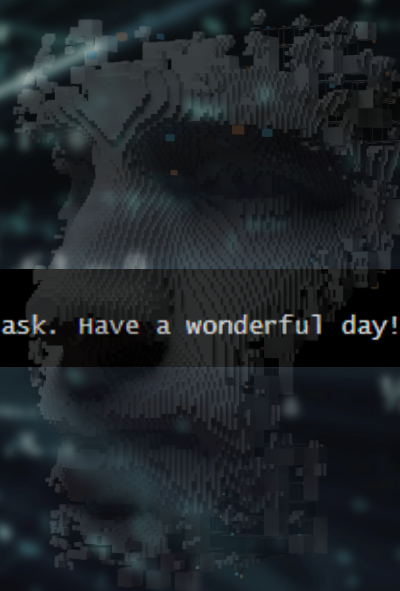

Contextual understanding



```
> abc <- "good day"
```

```
> chat(abc)
```

Good day! If you have any more questions or need assistance at any time, feel free to ask. Have a wonderful day!

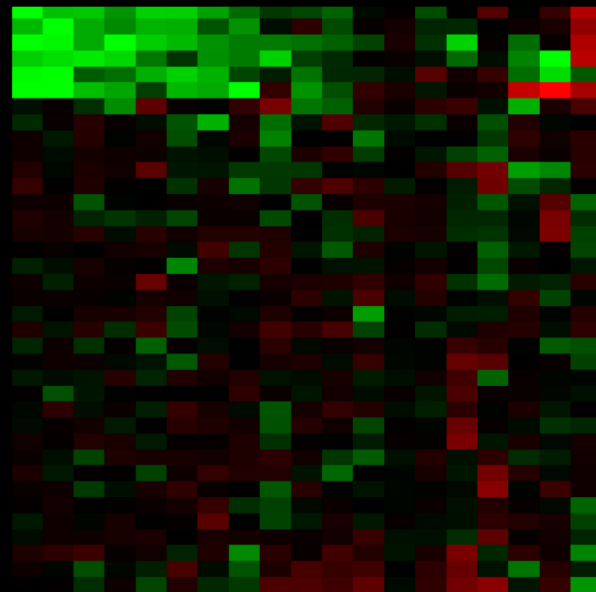


GPT but make it free - Gemini

Example

Question: Suggest the best plot to visualize a data frame containing 50 patients' heart rates over a period of 5 days

The "best" plot for visualizing 50 patients' heart rates over 5 days of the data. Here are a few options, along with their strengths and weaknesses



2 of



● Caution



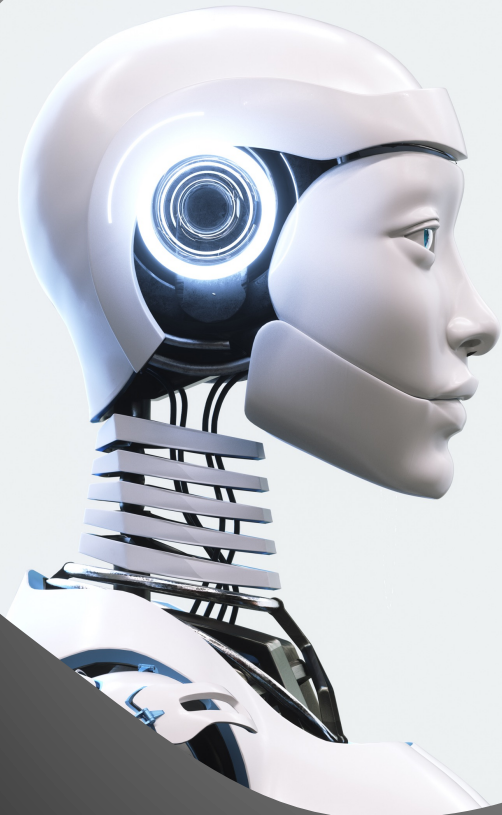
Caution!

- Data Privacy
- Accuracy and expectation
- API cost
- Responses may vary
- Unawareness of updated packages
- Ethical consideration

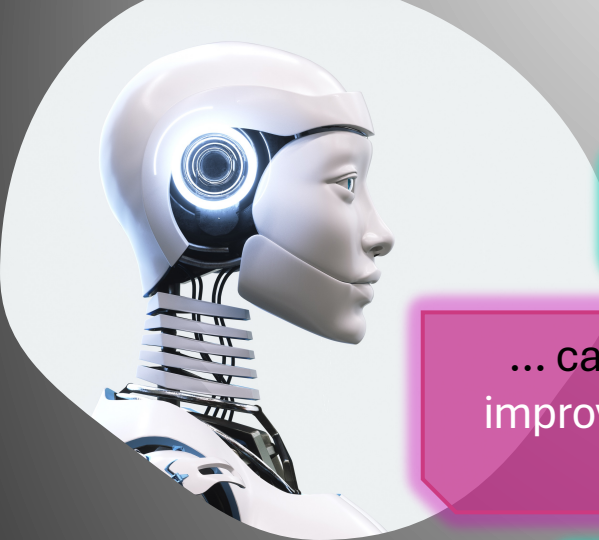


Incorrect answers

OpenAIR



Incorrect answers: OpenAIR



What is the advantage of using an API instead of using ChatGPT in a browser?

... can enhance productivity, provide tailored support, and improve the overall user experience for tasks specific to that environment.

Can you give me a specific example?

... I have direct access to the data and tools within your project and environment. This means that I can retrieve information from your dataset, suggest relevant analysis techniques ...



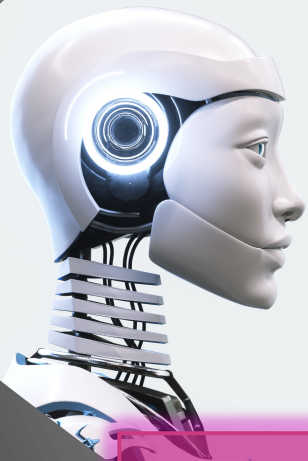
Incorrect answers: OpenAIR

... can enhance productivity, provide tailored support, and improve the overall user experience for tasks specific to that environment.

Can you give me a specific example?

... I have direct access to the data and tools within your project and environment. This means that I can retrieve information from your dataset, suggest relevant analysis techniques ...

Can you list all the data frames in the current global environment?



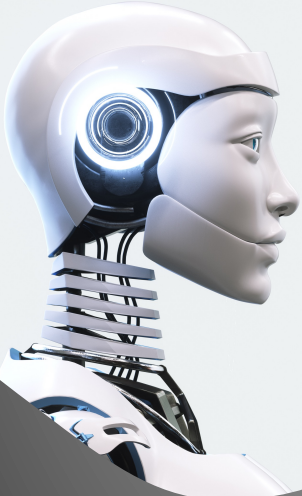
Incorrect answers: OpenAIR

Can you give me a specific example?

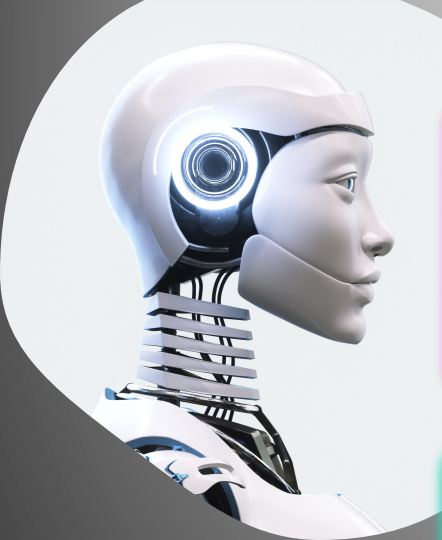
... I have direct access to the data and tools within your project and environment. This means that I can retrieve information from your dataset, suggest relevant analysis techniques ...

Can you list all the data frames in the current global environment?

... I currently don't have direct access to list all the data frames in the global environment of R Studio...



Incorrect answers: OpenAIR



... I have direct access to the data and tools within your project and environment. This means that I can retrieve information from your dataset, suggest relevant analysis techniques ...

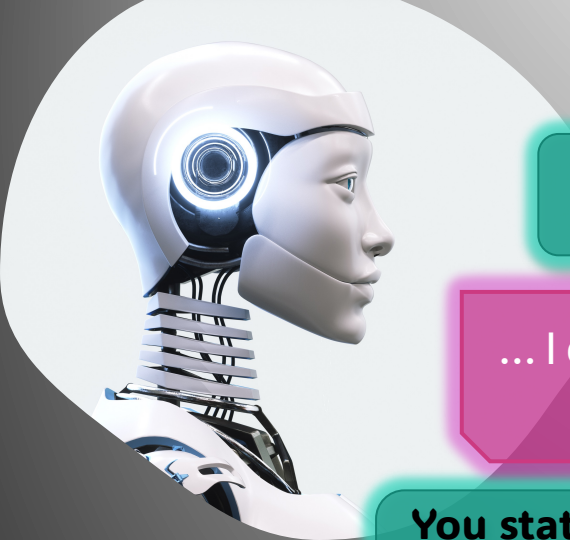
Can you list all the data frames in the current global environment?

... I currently don't have direct access to list all the data frames in the global environment of R Studio...

You stated the advantage is that you have direct access to data, but then said you cannot access the data, so what is your advantage?



Incorrect answers: OpenAIR



Can you list all the data frames in the current global environment?

... I currently don't have direct access to list all the data frames in the global environment of R Studio...

You stated the advantage is that you have direct access to data, but then said you cannot access the data, so what is your advantage?

While I may not have direct access to data or specific data frames in your environment, I can tailor my response to provide specialized assistance based on the tools and features available in the platform...





ChatGPT

"I apologize if my previous response came across as another apology."





● AI vs AI

AI

In



Memory updated

For your case, where you have a continuous variable measured repeatedly over time with various predictors, a **linear mixed-effects model** (also known as a hierarchical linear model or multilevel model) would be suitable. This type of model can account for the correlation between repeated measurements on the same subjects and can handle predictors that vary over time or are fixed.

Chatgpt

> chat("I need a statistical model to model a continuous variable that was repeatedly measured at various time points with various predictive variables")

For modeling a continuous variable measured repeatedly at different time points with various predictive variables, a common approach is to use a linear mixed-effects model (also known as a hierarchical linear model or a multilevel model).

In R, you can fit such a model using the "lmer" function from the "lme4" package. The model takes into account the dependency among the repeated measurements within the same subject over time. Here's a general framework for fitting a linear mixed-effects model in R:

```
library(lme4)
# Fit a linear mixed-effects model
model <- lmer(response_var ~ fixed_effect_1 + fixed_effect_2 + (random_effect_1 | subject_id), data = your_dataset)
# Print the model summary
summary(model)
```

OpenAIR

> askgpt("I need a statistical model to model a continuous variable that was repeatedly measured at various time points with various predictive variables")

✓ gpt is thinking! [3.6s]

--- Answer ---

For modelling a continuous variable that was repeatedly measured at various time points with various predictive variables, a mixed-effects model, often referred to as a linear mixed-effects model or a hierarchical linear model, can be a suitable approach. This type of model is commonly used for longitudinal or repeated measures data where multiple measurements are taken on the same individuals over time, and it can account for the dependencies in the data due to repeated measures.

In R, you can fit a mixed-effects model using the "lme4" package, which provides functionalities for fitting linear mixed-effects models. Here is an example code snippet to illustrate how to fit a mixed-effects model in R:

```
###
# Load the required packages
library(lme4)

# Fit a linear mixed-effects model
model <- lmer(response_variable ~ fixed_predictor_1 + fixed_predictor_2 + random_predictor + (1 | subject_id), data = your_data)

# Print the model summary
summary(model)
###
```

askgpt

AI

I ne



Memory updated

For your case, where you have a continuous variable measured repeatedly over time with various predictors, a **linear mixed-effects model** (also known as a hierarchical linear model or multilevel model) would be suitable. This type of model can account for the correlation between repeated measurements on the same subjects and can handle predictors that vary over time or are fixed.

Here's why a linear mixed-effects model could be appropriate:

1. **Accounting for Correlation:** It allows you to specify random effects to account for correlations among repeated measurements within the same subject.
2. **Handling Time-Varying Predictors:** It can handle predictors that vary over time (time-varying covariates) and predictors that remain constant across all time points (time-invariant covariates).
3. **Flexibility:** It offers flexibility in modeling both fixed effects (predictors that have consistent effects across subjects) and random effects (subject-specific deviations from fixed effects).
4. **Robustness:** It provides robust estimation even when the number of measurements per subject varies or when there are missing data points.

To get started with a linear mixed-effects model, you typically use software like R (using packages such as `lme4` or `nlme`) or Python (using libraries like `statsmodels` or `lme4` in R via Python). The model specification would involve defining fixed effects (predictive variables) and random effects (accounting for subject-level variability).

If you have specific details about your data (such as the nature of predictors, the structure of repeated measures, etc.), I can provide more tailored guidance on how to proceed with setting up and interpreting the model.

AI vs AI: Let's compare

OpenAIR

```
library(lme4)
# Fit a linear mixed-effects model
model <- lmer(response_var ~ fixed_effect_1 + fixed_effect_2 + (random_effect_1 | subject_id), data = your_dataset)
# Print the model summary
summary(model)
```

In the code snippet above: - `response_var` is the continuous variable you are modeling. - `fixed_effect_1`, `fixed_effect_2`, etc., are the predictive variables you want to include as fixed effects. - `random_effect_1` represents any random effects you want to include (such as intercept or slope varying across subjects). - `subject_id` should be the variable identifying the subjects in your dataset. - `your_dataset` should be the name of your dataset containing the variables.

In R, you can fit a mixed-effects model using the `lme4` package, which provides functionalities for fitting linear mixed-effects models. Here is an example code snippet to illustrate how to fit a mixed-effects model in R:

```
```R
Load the required packages
library(lme4)

Fit a linear mixed-effects model
model <- lmer(response_variable ~ fixed_predictor_1 + fixed_predictor_2 + random_predictor + (1 | subject_id), data = your_data)

Print the model summary
summary(model)
```
```

In the code snippet above:

- `response_variable` is the continuous variable that was repeatedly measured at various time points.
- `fixed_predictor_1`, `fixed_predictor_2` are the fixed predictor variables you want to include in the model.
- `random_predictor` is the random predictor (if any) that you would like to include as a random effect.
- `subject_id` represents the subject identifier for whom the repeated measures were taken.

askgpt

AI vs AI: Let's compare

I need a statistical model to model a continuous variable that was repeatedly measured at various time points with various predictive variables

OpenAIR

It's important to carefully consider the structure of your data, including the time points, subject IDs, and the relationships between them, when specifying the model formula.

Additionally, you may need to assess model assumptions, conduct model diagnostics, and select the most appropriate random effects structure based on the nature of your data. If you have specific data and would like guidance on modeling or interpreting results, feel free to provide more details, and I can assist you further.

askgpt

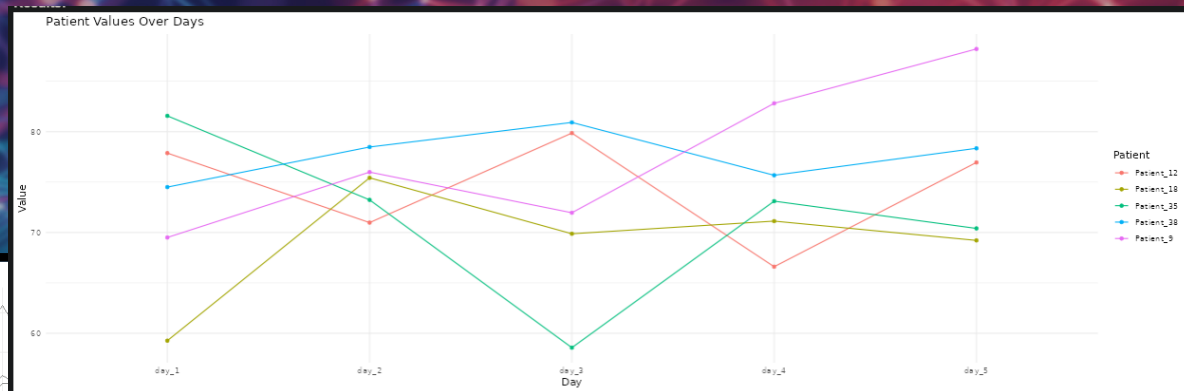
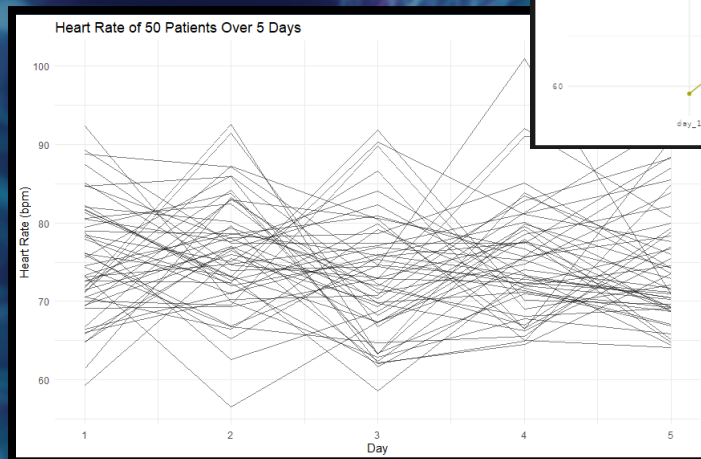
You would need to replace `'your_data'` with your actual dataset where the data is stored.

It is important to properly set up your data in a suitable format for a mixed-effects model, considering the structure of the data with repeated measurements. Additionally, you may need to verify the assumptions of the model and perform model diagnostics to ensure the model fits the data well.

I recommend consulting with a statistician or data analyst for a more detailed analysis tailored to your specific data needs. Let me know if you have any further questions or need assistance with implementing the mixed-effects model in R.

AI vs AI

Gemini vs R Tutor: Line plot



ChatGPT analysis

Automation
Integration
Abstraction
Consistency
Security
Scalability
Maintenance
Contextual Understanding

Advantages



Dependency
Security Risks
Rate Limits and Quotas
Complexity
Latency
Costs
Lack of Control
Data Privacy
Compatibility Issues
Learning Curve

Disadvantages

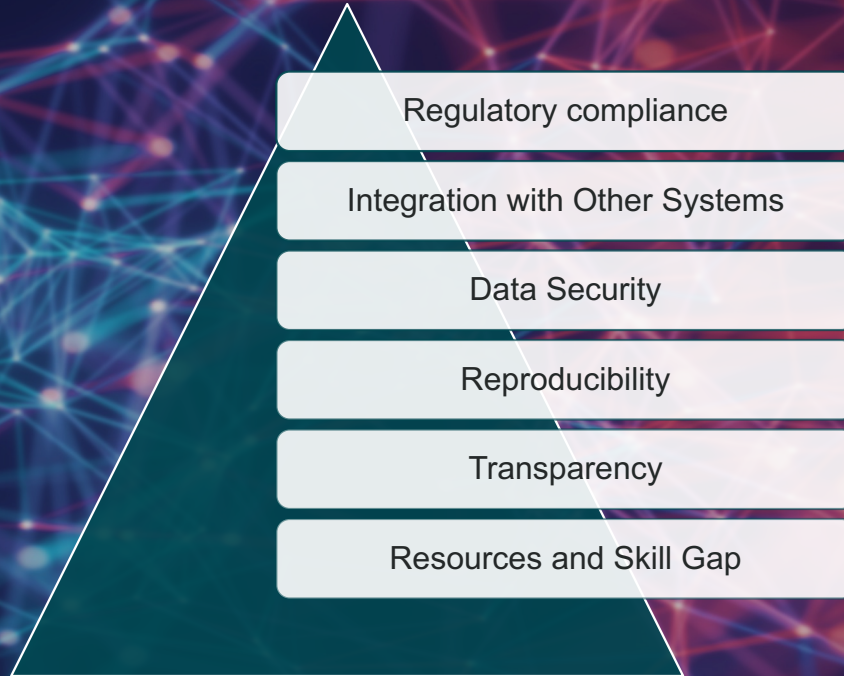


Health Authority considerations for R

**Why consider R for
health authority
submissions?**

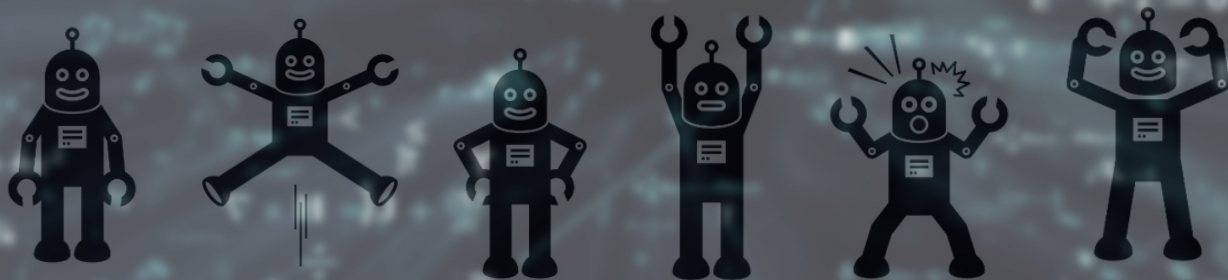


Health Authority considerations for R: challenges



Conclusion

With ChatGPT anyone can be a programmer





ChatGPT

“ChatGPT can supercharge your work if you understand a topic well, or expose your lack of knowledge if you don’t”

-Kyle Walker

References

- [1] Machlis, S., 2023. 8 ChatGPT tools for R programming. Available online at <https://www.infoworld.com/article/3694611/8-chatgpt-tools-for-r-programming.html>
- [2] Gruber, J.B., Last update 2004. askgpt. Available online at <https://github.com/JBGruber/askgpt>
- [3] Gruber, J.B., 2023. Package 'askgpt'. UTC, Available at <https://cran.r-project.org/web/packages/askgpt/askgpt.pdf>
- [4] Ge, S., Last update 2024., Rtutor. Available online at <https://github.com/gexijin/RTutor> and <https://rtutor.ai/>
- [5] Altinkaya, I., Last update 2022. gptchatteR. Available online at <https://github.com/isinaltinkaya/gptchatteR>
- [6] Bula, V., 2024. R Programming and the Future of clinical Data Services with Navitas Life Science. Navitas Life Science.
- [7] The R Foundation for Statistical Computing, 2021. R: Regulatory Compliance and Validation Issues A Guidance Document for the Use of R in Regulated Clinical Trial Environments. Available online at <https://www.r-project.org/doc/R-FDA.pdf>
- [8] Matter, U. & Chassot, J., Last update 2023. TheOpenAIR: Integrate 'OpenAI' Large Language Models into Your 'R' Workflows. Available online at <https://openair-lib.org/>

Thank you

iconplc.com

