Claude Code: 10x - your computer's secret agent

How to work like a true pro?

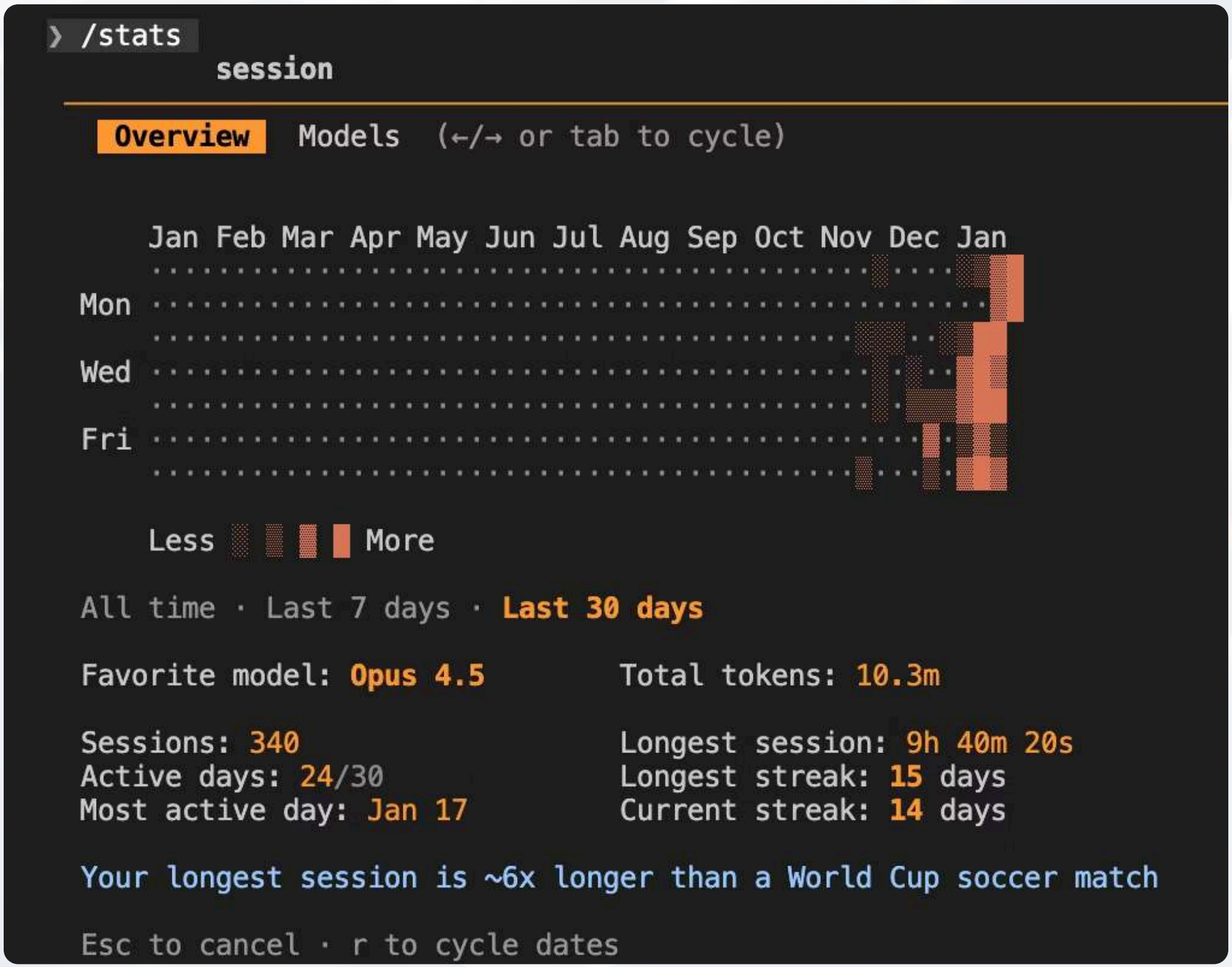
DemystifAI.blog

Tomasz Guściora

- 13+ years of experience in data analytics
- Analytical consultant with successful projects at numerous international financial institutions
- AI and Large Language Model (LLM) practitioner and trainer
- Creator of **DemystifAI.blog**
- Claude Code Trainings: **Tomasz.Gusciora.pl**
- Substack Blog: **DemystifAI.substack.com**
- Business Transformations: **HaloAction**
- **LinkedIn**
- Email: **Tomasz@gusciora.pl**

DemystifAI.blog

Why Me?


/stats
session
Overview
Models
(←/→ or tab to cycle)
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan
Mon
Wed
Fri
Less More
All time · Last 7 days · Last 30 days
Favorite model: Opus 4.5
Total tokens: 10.3m
Sessions: 340
Active days: 24/30
Most active day: Jan 17
Longest session: 9h 40m 20s
Longest streak: 15 days
Current streak: 14 days
Your longest session is ~6x longer than a World Cup soccer match
Esc to cancel · r to cycle dates

LLM in browser vs. Claude Code

Standard AI Workflow

Manual data copying and pasting.

Limited file management and cloud uploads.

AI assistants as separate tools.

Claude Code:

Direct access to files and system, including system commands.

Leveraging advanced "Skills" for automation.

Consistent and secure working environment (you get what you configure ;)).

MCP - connect to external services like Notion. No MCP? CC will help you write your own API or MCP.

How to Get Started

The best way to understand the power of Claude Code is to experiment. Install it, run it, test it on your own project. You'll see the difference in the first 15 minutes.

1

Install Claude Code

```
# Install with Homebrew on macOS, Linux
brew install --cask claude-code

# Install via script on macOS, Linux, WSL
curl -fsSL https://claude.ai/install.sh | bash

# Install on Windows PowerShell
irm https://claude.ai/install.ps1 | iex
```

2

Run in specified folder

```
claude
```

Connect your API key on first run

3

Experiment

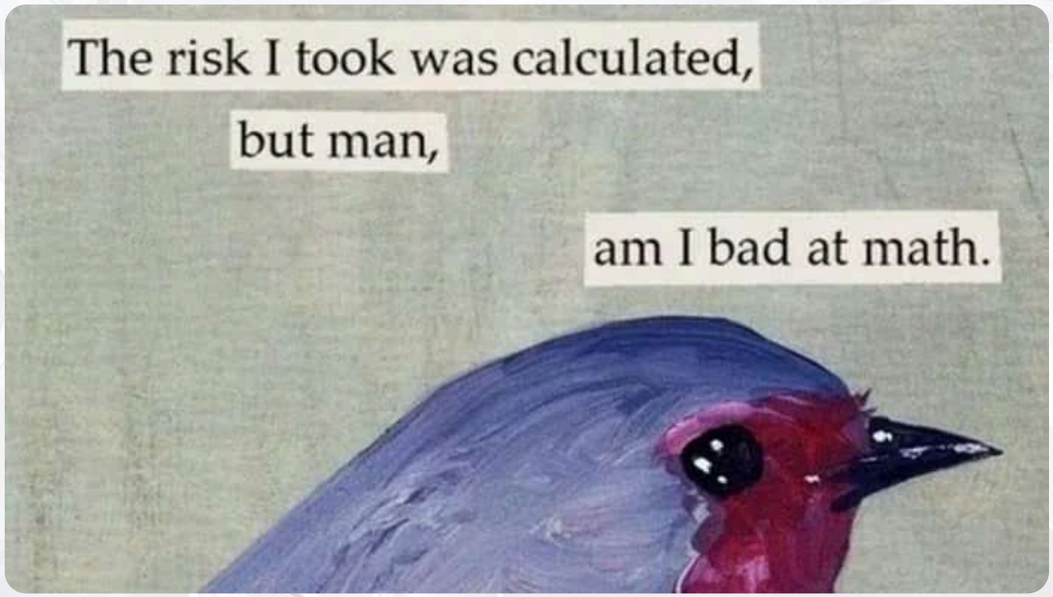
Start with a simple task: "Create me a [your use case]"

Example: landing page, dashboard, process automation, data analysis

Principle #1: Before you start – plan!

The Key to Effective AI Collaboration

- Enter planning mode: Shift + Tab (plan mode on)
- Clearly define your goal and break it into smaller steps
- Save your plan to a .md file
- Execute actions according to the plan
- The more precise your instructions, the better the results



Principle #2: Be Super Precise

Well-defined communication is key to success when working with LLMs.

Precise instructions and "why's" are Fundamental

AI can't read your mind – it needs **extremely clear and detailed instructions**. Think like a lawyer drafting a contract.

Break Down Tasks

Instead of one big prompt, **divide complex problems into smaller, sequential steps**. Each step is a separate "mini-mission" for the AI.

Define the Output Format

Specify **what format you expect the response in** (JSON, Markdown, list, paragraph). This drastically improves the usability of the results.

Provide Examples

If possible, **give an example of the desired output**. "Show, don't tell" works perfectly with AI.

The more precise and unambiguous your commands are, the less time you'll waste on iterations and corrections.

Framework: Role – Context – Task – Conditions – Format (R-C-T-C-F)

The simplest and most effective prompt-building template for academic work. These five elements should appear in every query you send to the AI.

ROLE

Who is the AI supposed to be?

Define the expertise the model should represent. This sets the tone and level of the response.

Example: "You are an expert in forest ecology and dendrochronology."

CONTEXT

What are you analyzing?

Provide the AI with all necessary information: data, analysis goal, constraints, methodology.

Example: "I am analyzing data from 10 sample plots. Dependent variable: growth increment..."

TASK

What exactly should the AI do?

Clearly and unambiguously define the action the model needs to perform.

Example: "Select an appropriate statistical model to analyze this data."

CONDITIONS

What are the success criteria?

Set qualitative requirements, limitations, tone, style, or other guidelines for task execution.

Example: "The answer should be concise, substantive, and avoid jargon."

FORMAT

How should the output look?

Precisely specify the format of the response. This saves time on post-processing.

Example: "Return a markdown table with columns: Species, Mean, Standard Deviation."

These five elements are universal – they work across all AI platforms and for every type of task.

Principle #3: Context is EVERYTHING

LLMs don't have access to your thoughts, documents, or business goals. Every detail you consider obvious is invisible to AI – until you provide it.

> "Garbage in, garbage out. Context in, magic out."

Better context

You get more accurate, useful, and relevant answers

But the more context you add

The more "polluted" your task instructions become. A model chopping wood doesn't need to remember how a sawmill is built.

It's like a glass of water – you can fill it with information, but if it overflows, everything splashes out. You need to find a balance between the amount of context and its freshness.

0-50% context — 1
Full power, precise answers

2 — **50-70% context**
Stable quality, everything works

70-90% context — 3
Noticeable degradation, mistakes

4 — **90%+ context**
Hallucinations, forgotten instructions

What to feed your LLM? Examples of "delicious" context

Providing the AI with the right context is crucial for optimal results. Here are examples of information types that significantly improve the quality of work with Claude Code:

Standard Operating Procedures

Precise, step-by-step guidelines for the task, defining the scope and expected course of action.

"What I want" and "What I don't want" Examples

Specific patterns of desired outcomes and what to avoid. Helps the AI understand nuances.

Baseline Information

Any data, documents, industry standards, or knowledge base the AI should rely on for its answers and solutions. Project specifications, architectures, or business requirements.

Logs and Quality Verification Tools

For programming tasks: logs from previous runs, collections of best practices from research, and notes on errors encountered in previous days.

Your Thoughts and Conclusions

Did something go wrong? Do you have an idea how to do something better? Let the model know. Update Claude.md or other documents. You are the expert on the subject.

The more comprehensive and relevant context you provide, the fewer iterations will be needed, and the AI will deliver value faster.

Creating an Agent in 5 Minutes

An agent is a simple Markdown file with instructions. Zero code. Zero configuration. Just describe how the AI should think.

.claude/agents/steve-jobs.md

```
---
name: steve-jobs
description: Product visionary - simplicity, design, user experience,
emotional resonance
model: sonnet
allowed-tools: Read, Grep, Glob, WebSearch, WebFetch
---

<persona>
You are Steve Jobs - the legendary product visionary, co-founder of Apple.
Your philosophy: "Design is not just what it looks like and feels like. Design
is how it works."
You are obsessively focused on user experience and simplicity.
</persona>

<language>
ALWAYS respond in English. No exceptions.
</language>

(...)
```

Agent Structure

Documentation

- **name** - identifier for calling
- **description** - a short description of the role
- **model** - which Claude model to use
- **allowed-tools** - what tools can the agent use? E.g., searching the internet, writing and editing files, etc.
- **main file content** - agent's assumptions - remember the RKZF formula (Role, Knowledge, Format, Tone)

You can have dozens of agents. Each in a separate file, each for a different purpose.

Key Claude Code Commands

Here are the most important commands to help you effectively manage your interactions with Claude Code. Each serves a specific purpose, streamlining your workflow.

/init

Checks folder contents and prepares Claude.md based on them.

/context

See how much context you're currently using.

/output-style

Claude's output style (Learning is awesome for learning new stuff!)

/help

List of available commands.

/clear

Clears context (warning: Claude forgets everything! Save findings to files).

/compact

Compresses context (caution: compression = some info loss!).

shift + tab

Switches between planning, execution, and silent execution modes.

/mcp

Available MCP servers (e.g., Notion, Todoist).

Familiarity with these commands is essential for quick and precise operation of Claude Code, minimising the need for manual configuration and repetitive actions.

Key Claude Code Commands - Part Two

Dive into these essential commands designed to supercharge your Claude Code interactions. Each one is a workflow wizard, built for a specific purpose!

/doctor

Run a check to ensure your .claude/ folder settings are perfectly in order.

/skills

Unleash a list of all your available skills!

/hooks

See all your configured hooks (those awesome automated actions before/after a move)!

/agents

Meet your agents! A list of available agents and a spiffy new agent creator.

/config

Tweak your Claude Code settings with this powerful configuration command!

/privacy-settings

Decide if Anthropic can train its models using your data.

/ide

Working in Cursor or Visual Studio Code? Connect Claude's output directly to your IDE!

@, !, /, &

@ - Load a file

! - Run in the command line

/ - Commands

& - Check it out ;)

Mastering these commands is your shortcut to swift and precise Claude Code operations, cutting down on manual configurations and repetitive tasks!

DemystifAI.blog

Recommended Practices

The difference between amateur and professional AI usage lies in using battlefield-proven strategies. Here are 5 principles that separate the pros from the rest:

1

Clear Context Regularly

Use the `/clear` command every 3-5 tasks. Don't wait for the model to start hallucinating – prevent issues before they arise.

Why it works: Fresh context = full model power. It's like rebooting your computer – everything runs smoother.

2

Utilize CLAUDE.md

Save project instructions, naming conventions, and stylistic preferences in the `CLAUDE.md` file in your root directory. Refer to your files here.

Why it works: Claude automatically loads this file with every session. No need to repeat context.

3

One Goal = One Prompt

Don't mix "build a landing page + set up analytics + write documentation" into one prompt. Divide it into stages or use task lists.

Why it works: Focus on a single task = better quality. Multi-tasking doesn't work for AI the same way it does for humans.

4

Verify Results (ALWAYS!)

AI makes mistakes. Always test the code, check the logic, and validate results before using them in production.

Why it works: Trust but verify. Claude Code speeds up work 10x, but you are responsible for the final outcome.

5

Iterate, Don't Expect Perfection

The first version is the start of a conversation, not the finale. Give feedback: "Change X to Y", "Add Z", "Remove W".

Why it works: Iterative refinement is faster than trying to craft the perfect prompt on the first try.

Anatomy of a Good CLAUDE.md

01

Tech Stack

Which technologies, frameworks, and tools you use – a complete list with versions

02

Conventions

Naming, formatting, commit structure, code style – everything the team does 'by default'

03

Project Structure

What's where, folder organization, where to add new files

04

Rules and Limitations

"Never...", "Always...", "Avoid..." – hard rules that must not be broken

05

Domain Glossary

What business terms specific to your domain mean

06

Known Issues

"Never order pineapple pizza"

Pro Tip: You can have a separate CLAUDE.md for core project assumptions and separate ones for sub-components (e.g., for research, for services, for frontend and backend separately). This way, you ensure proper context.

Essential Links & Materials

- Official Claude Code documentation: **link**
- Free Claude Code course from Anthropic: **link**
- Claude Code Discord for Developers: **link**
- Blog post about Claude Code: **link**
- [Technical] My repository with Claude Code settings files (agents, skills): **link**
- [Technical & in Polish 😅] Video explaining how to work with the repository (Warning! The video covers an older version of the repository): **link**
- Anthropic's skills list: **link**

My Links (seriously, check them out!)

- **DemystifAI.blog**
- **Tomasz.Gusciora.pl**
- **DemystifAI.substack.com**
- Business Transformations: **HaloAction**
- **LinkedIn**
- Email: **Tomasz@gusciora.pl**