



# Git ile versiyon kontrolü

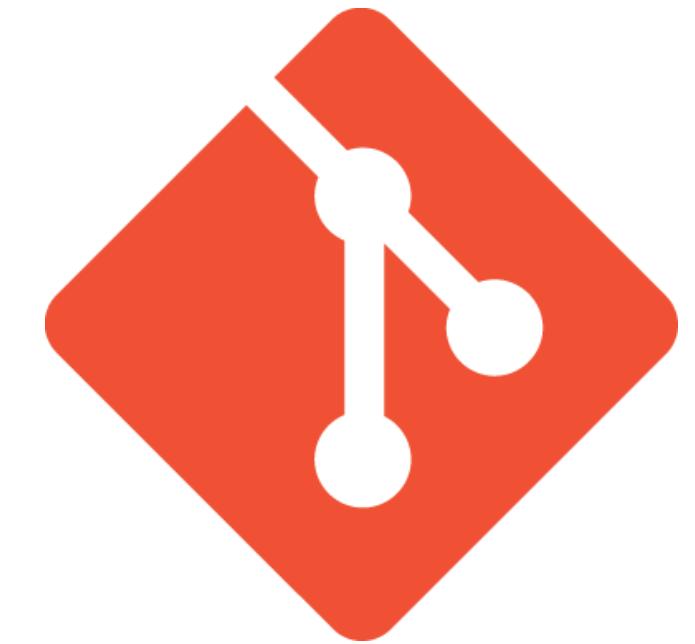
## araştırmacılar için bir giriş

Ali Seyhun Saral  
Max Planck Enstitüsü



# git nedir?

*"Dosyalar üzerinde yapılan değişiklikleri takip etmeye yarayan bir yazılım"*



# git nedir?

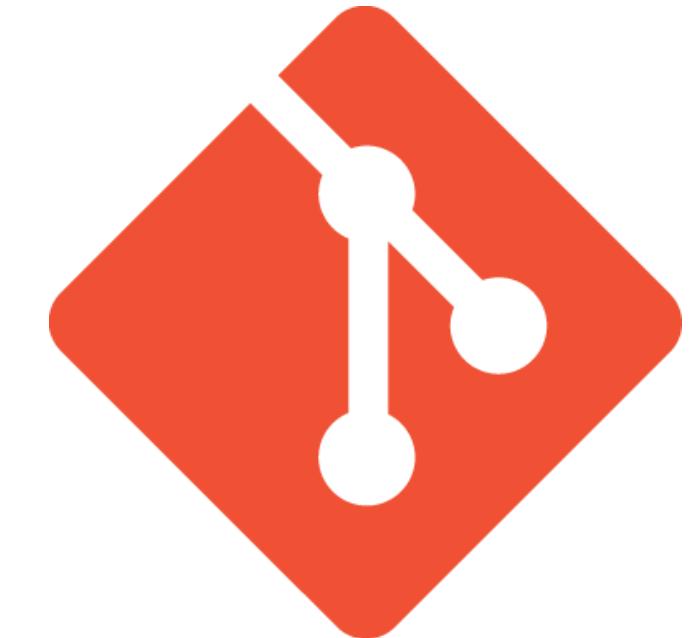
*"Dosyalar üzerinde yapılan değişiklikleri takip etmeye yarayan bir yazılım"*



- En popüler versiyon kontrol yazılımı
- Açık-kaynaklı ve bedava

# git nedir?

*"Dosyalar üzerinde yapılan değişiklikleri takip etmeye yarayan bir yazılım"*



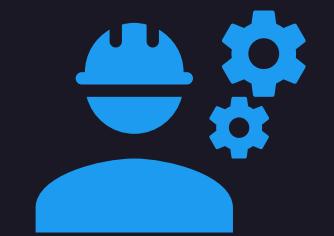
- En popüler versiyon kontrol yazılımı
- Açık-kaynaklı ve bedava

***Yazılım ve veri analizi projeleri için de facto bir proje standartı.***

# Kimler kullanıyor?



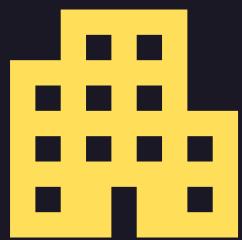
yazılımcılar



mühendisler



bilim insanları



kamu  
kuruluşları



teknoloji

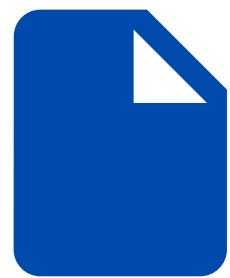


yazarlar

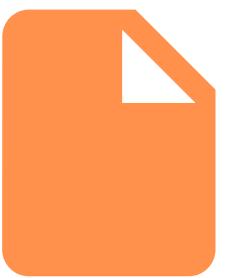
# ne işe yarar?



analiz.R



analiz2.R



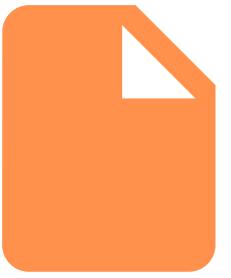
data1\_edit.xls



analiz\_eski  
.R



analiz2021  
ekim.R



data1.xls



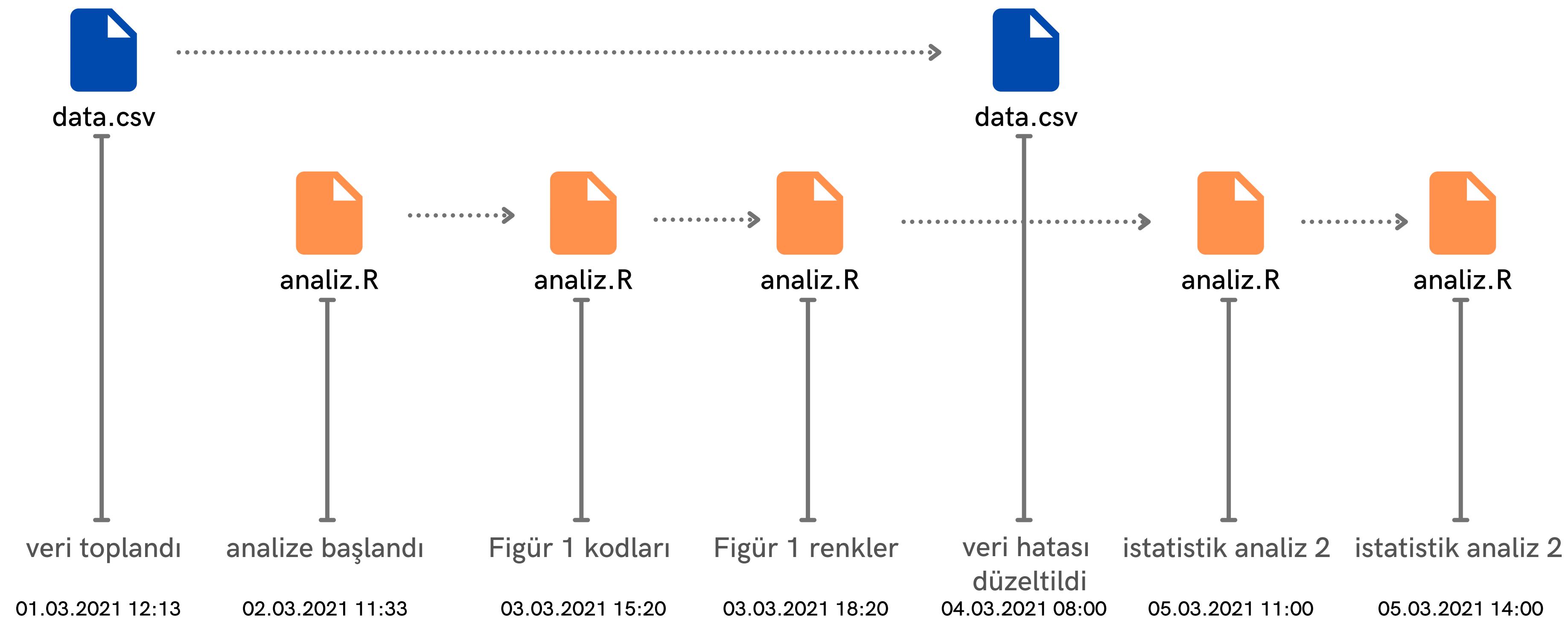
analiz2021  
ekim\_son.

R



1

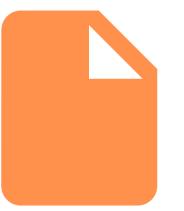
# zaman makinesi ve günlüğü





2

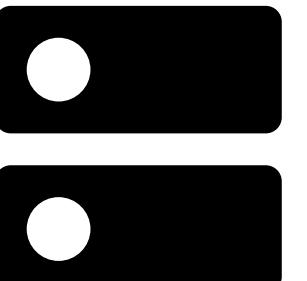
# ortak çalışma yöneticisi



**gitar.txt**



**sozler.txt**



**davul.txt**



**bas.txt**



sozler.txt



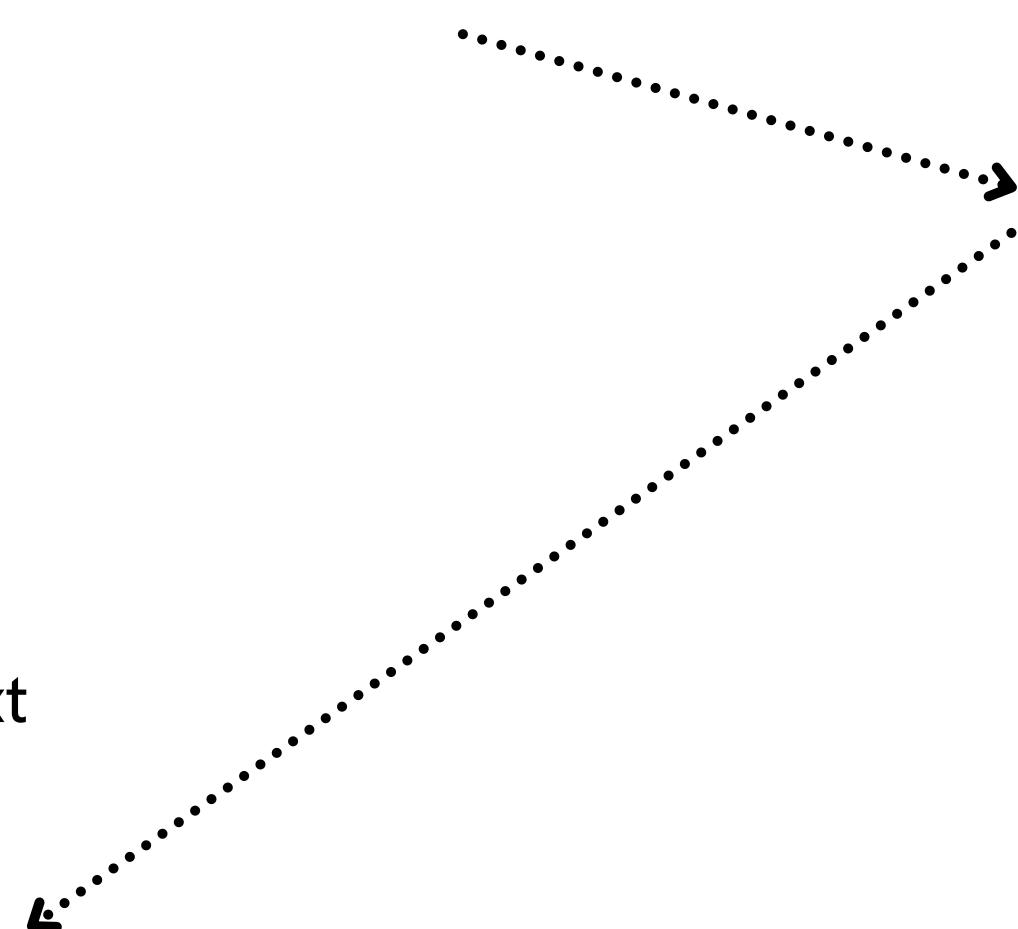
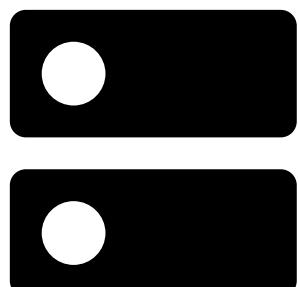
gitar.txt



gitar.txt



sozler.txt



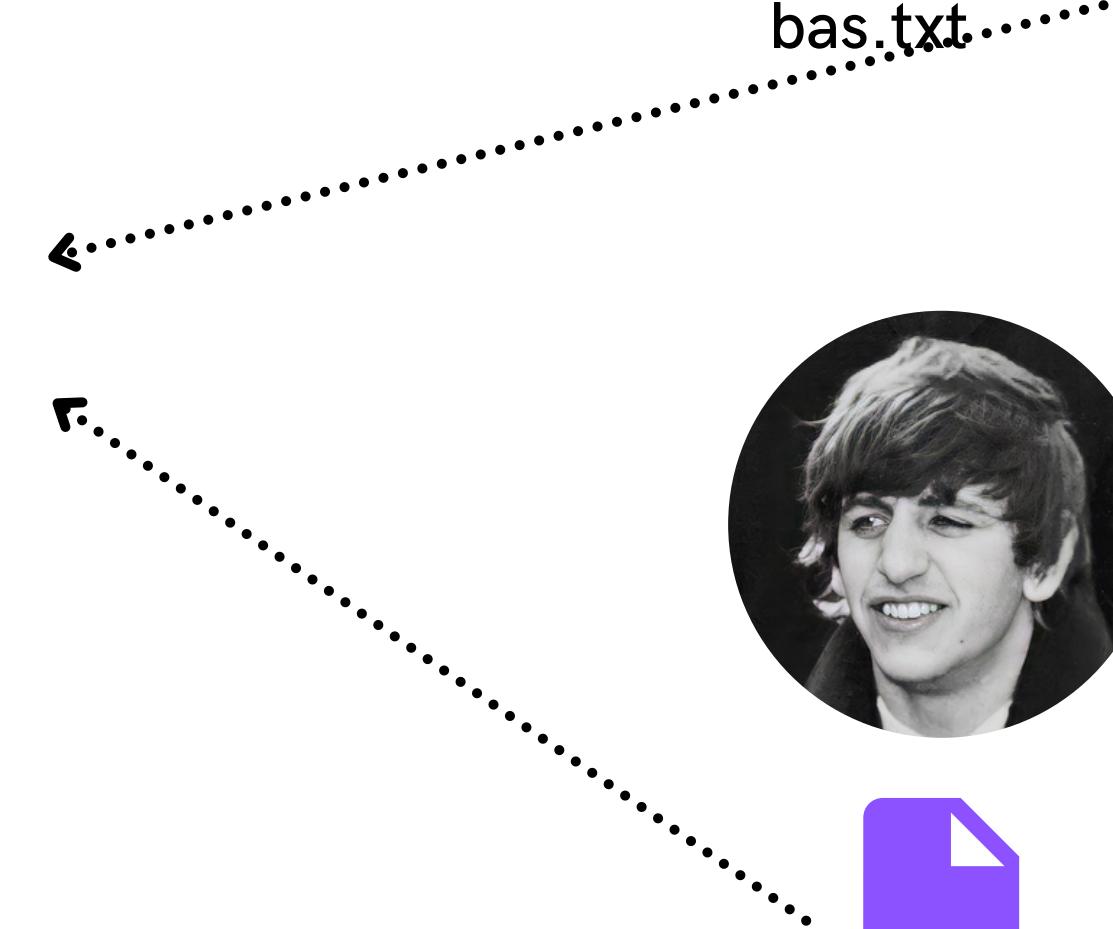
davul.txt



bas.txt



sozler.txt





sozler.txt



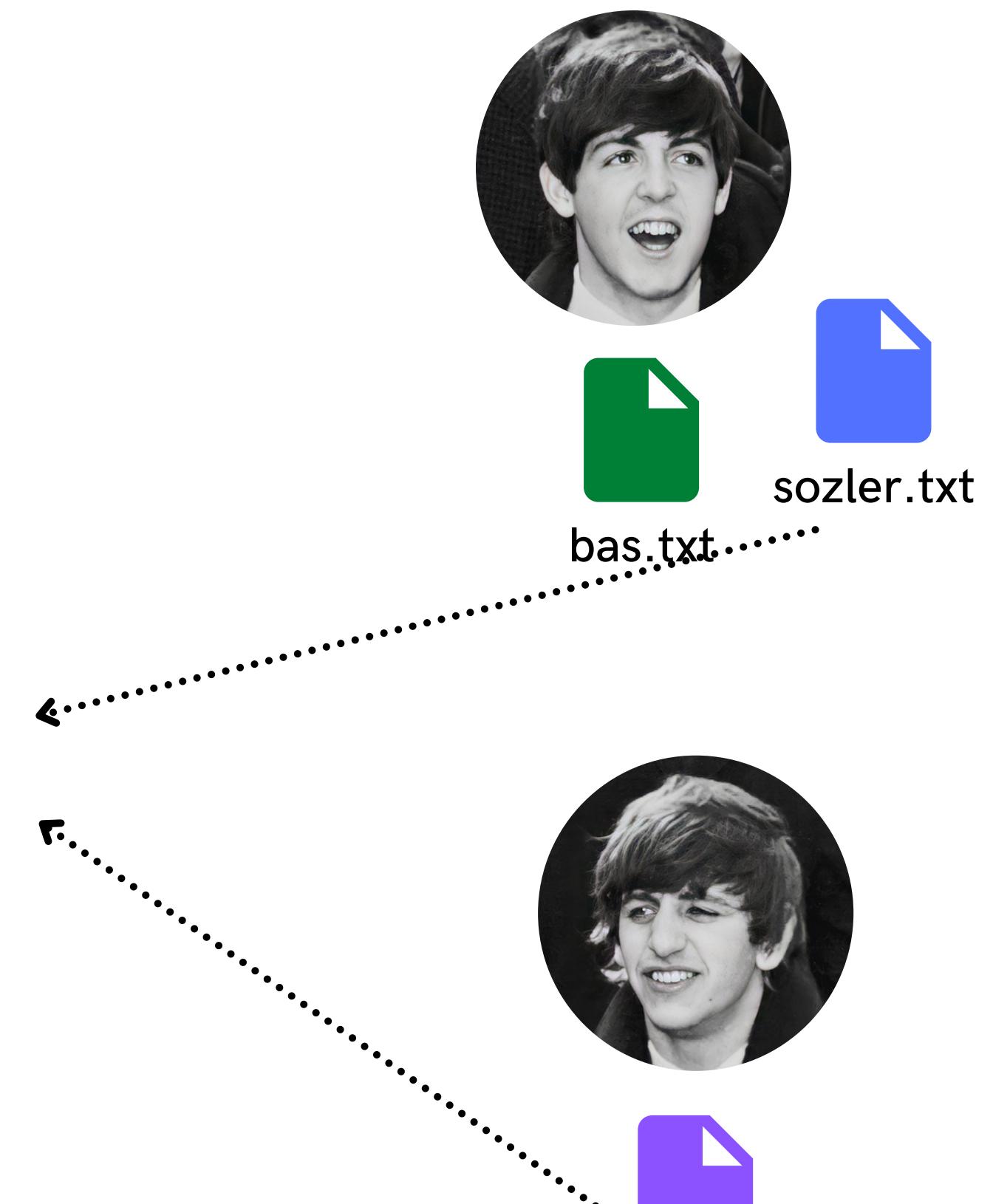
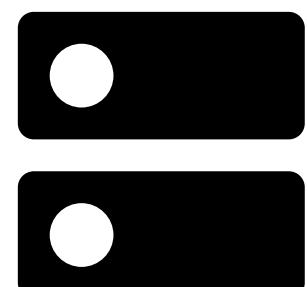
gitar.txt



gitar.txt



sozler.txt



bas.txt



sozler.txt



davul.txt





sozler.txt



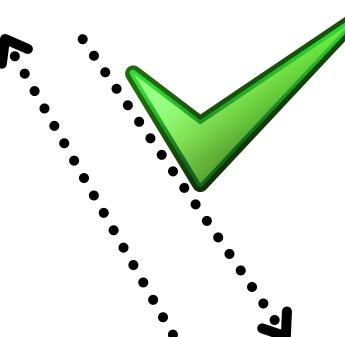
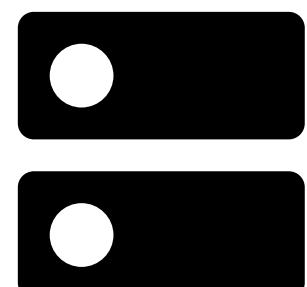
gitar.txt



gitar.txt



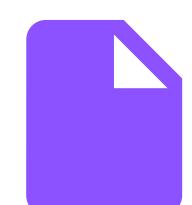
sozler.txt



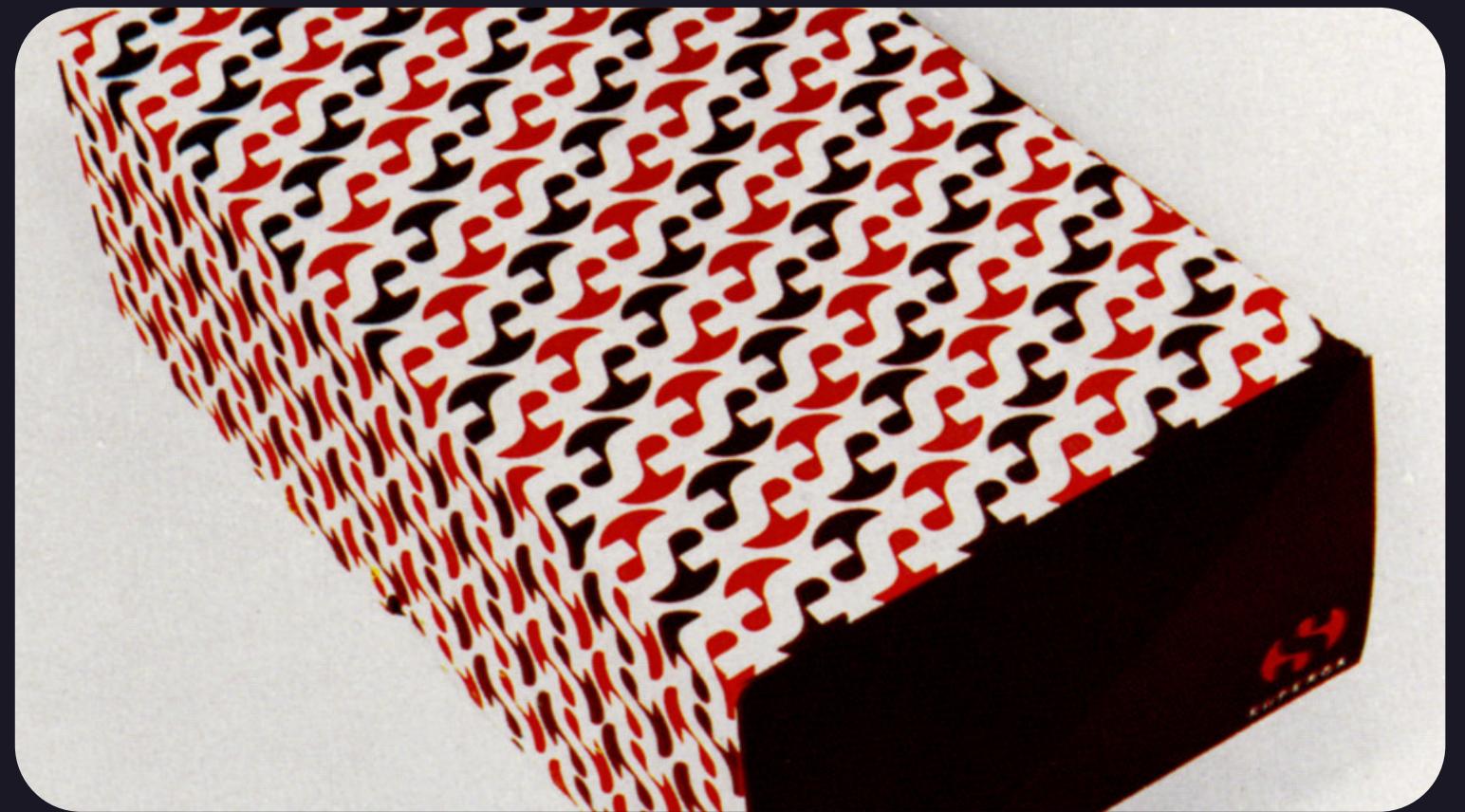
bas.txt



sozler.txt



davul.txt



3

# proje paketleme

master ▾

1 branch

2 tags

Go to file

Add file ▾

Code ▾



## About

Materials for "The Stability of Conditional Cooperation; (Andreozzi, Ploner, Saral; Scientific Reports, 2020)

DOI [doi.org/10.1038/s41598-020-70681-z](https://doi.org/10.1038/s41598-020-70681-z)

Readme

View license

## Releases 2

code and data 20200726 Latest  
on 26 Jul 2020

+ 1 release

## Packages

No packages published  
[Publish your first package](#)

## Languages

R 100.0%

seyhunsaral Update LICENSE

979abf8 on 31 Aug 2020 50 commits

code	some comments and fix table fig names	10 months ago
data	initial commit	2 years ago
figs	initial commit	2 years ago
ztree	add z-tree files	2 years ago
.here	initial commit	2 years ago
LICENSE	Update LICENSE	9 months ago
README.md	Update README.md	10 months ago
install.R	Create install.R	12 months ago
runtime.txt	Create runtime.txt	12 months ago

README.md

License MIT License CC BY 4.0 DOI [10.5281/zenodo.3960623](https://doi.org/10.5281/zenodo.3960623)

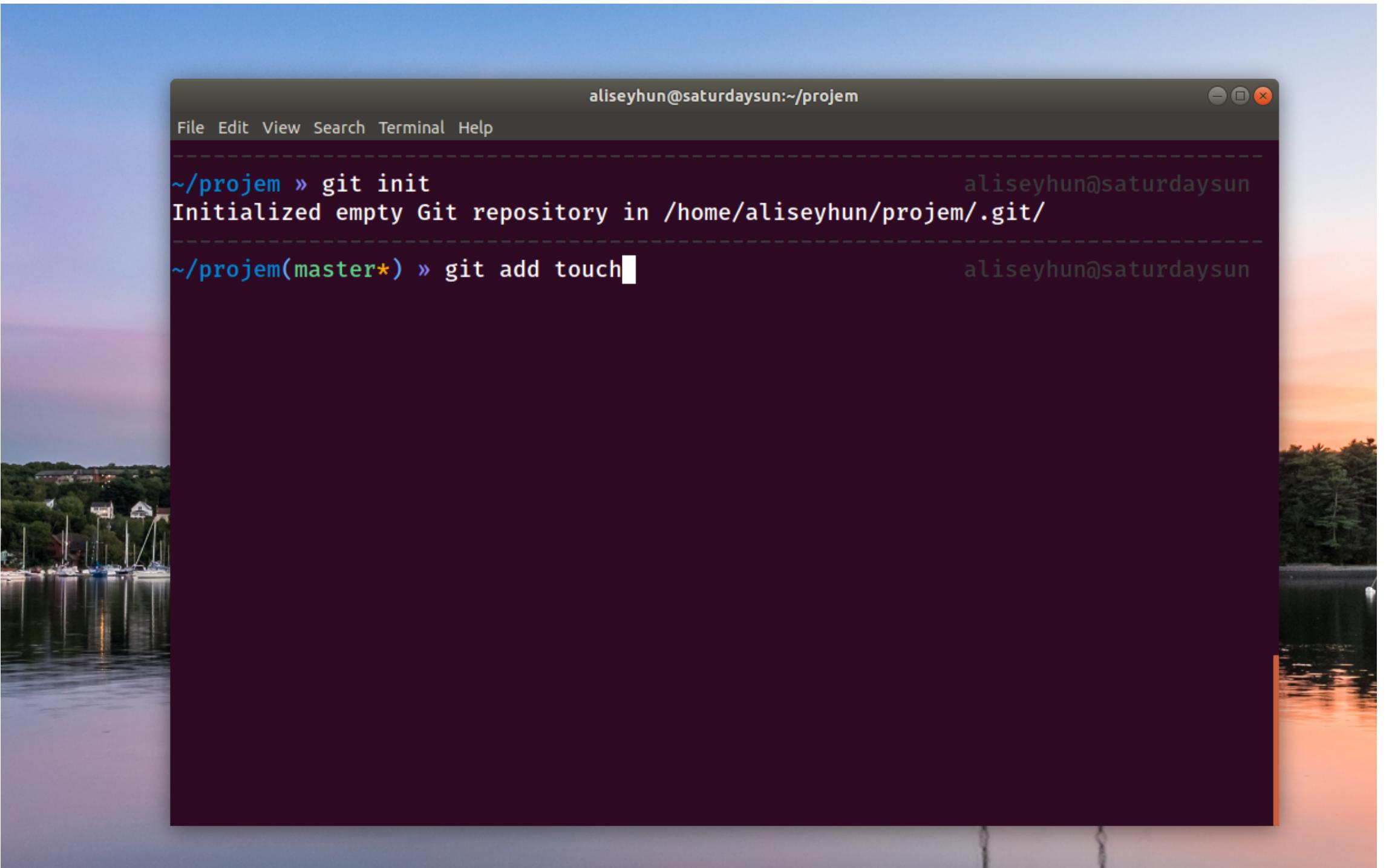
# The Stability of Conditional Cooperation (Andreozzi, Ploner, Saral, 2020)

# Git'i kullanmak

Git programını Windows,  
MacOS ve Linux tabanlı işletim  
sisteminize kurabilirsiniz.  
<https://git-scm.com/downloads>

Git ile etkileşim komut  
satırı/terminal üzerinden  
gerçekleşir.

Standart bir kullanım için sınırlı  
sayıda komut öğrenmek  
yeterlidir.



```
aliseyhun@saturdaysun:~/projem
File Edit View Search Terminal Help
~/projem » git init
Initialized empty Git repository in /home/aliseyhun/.projem/.git/
~/projem(master*) » git add touch
```

# Git nasıl çalışır?



data.csv



analiz.R



tasarım.png

# Git nasıl çalışır?



data.csv



analiz.R



tasarım.png

git init

Mevcut klasörde bir git takibi başlatma

# Git nasıl çalışır?



.git



data.csv



analiz.R



tasarım.png

git init

Mevcut klasörde bir git takibi başlatma

# Git nasıl çalışır?

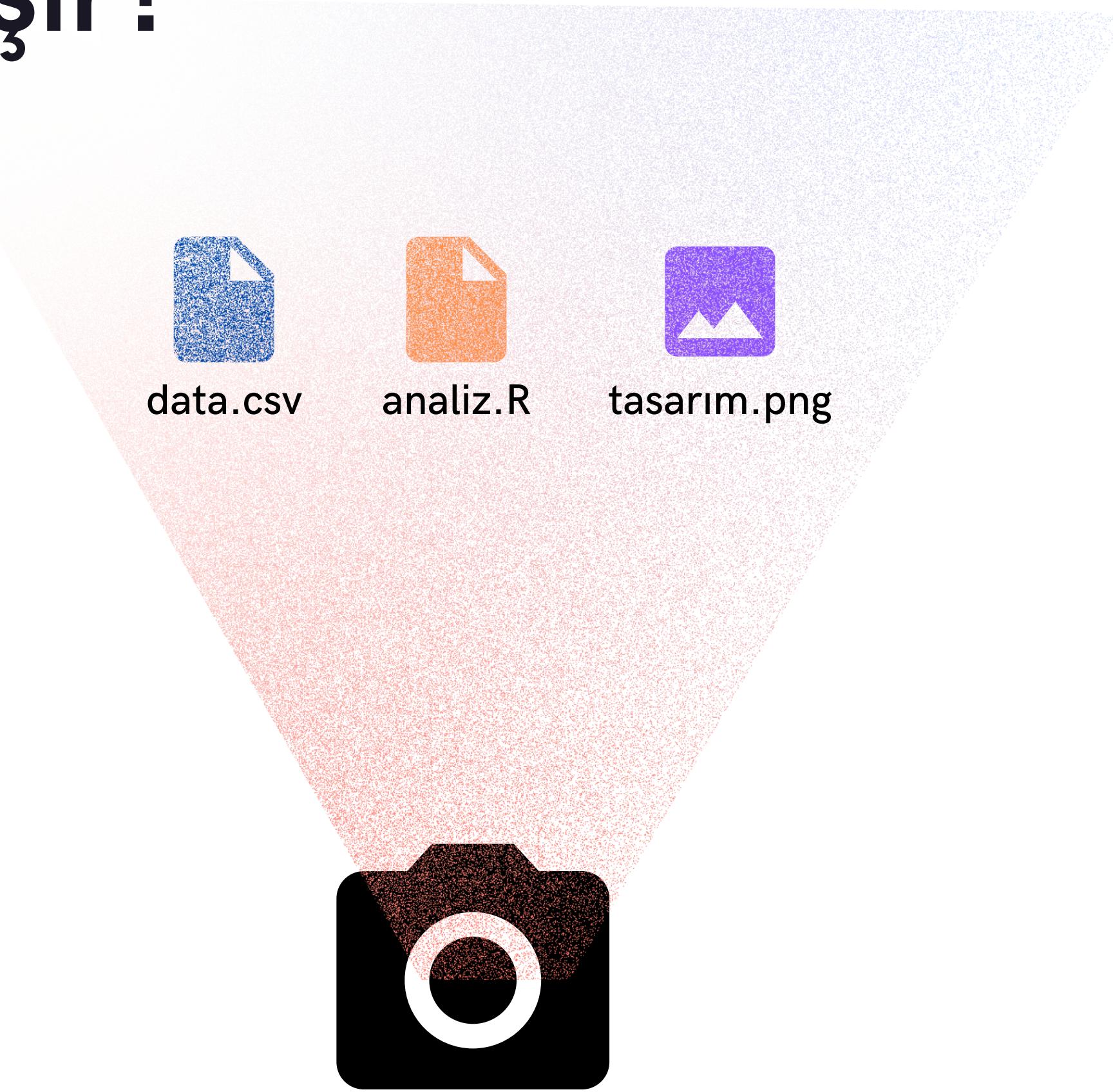


- Versiyonlarla ilgili veriler .git klasöründe tutulur
- Komutlarla takip edilmek istenen dosya ve klasörler eklenir
- Değişiklikler manuel olarak eklenir

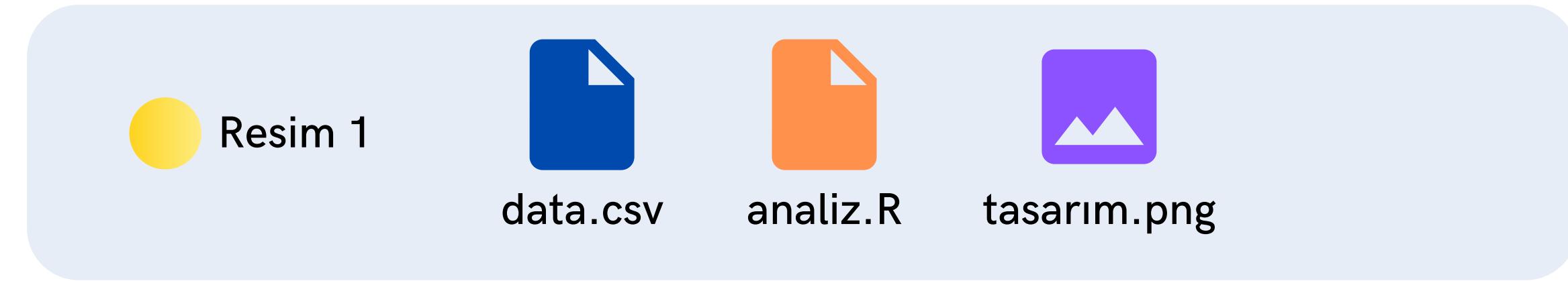
`git init`

Mevcut klasörde bir git takibi başlatma

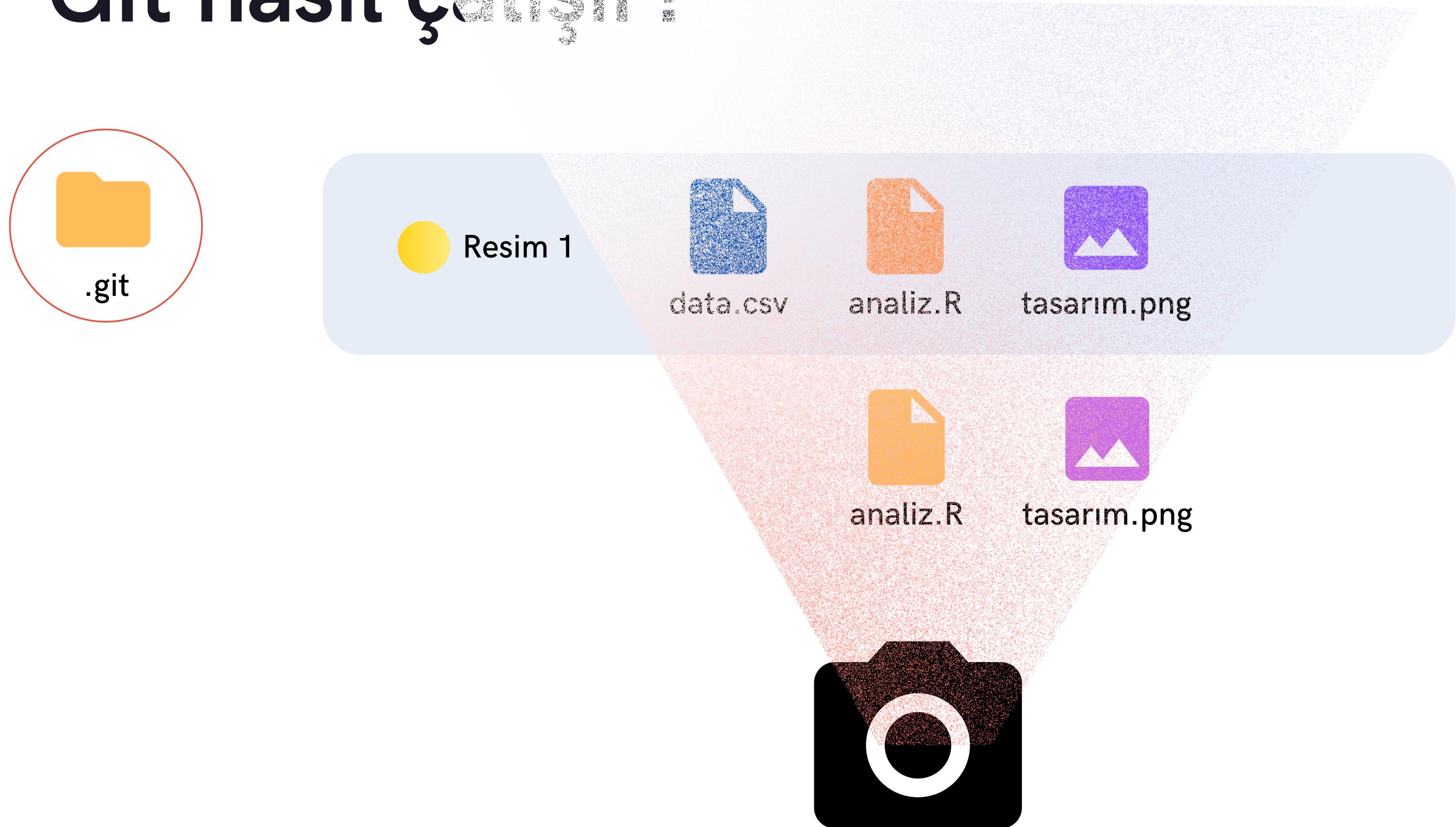
# Git nasıl çalışır?



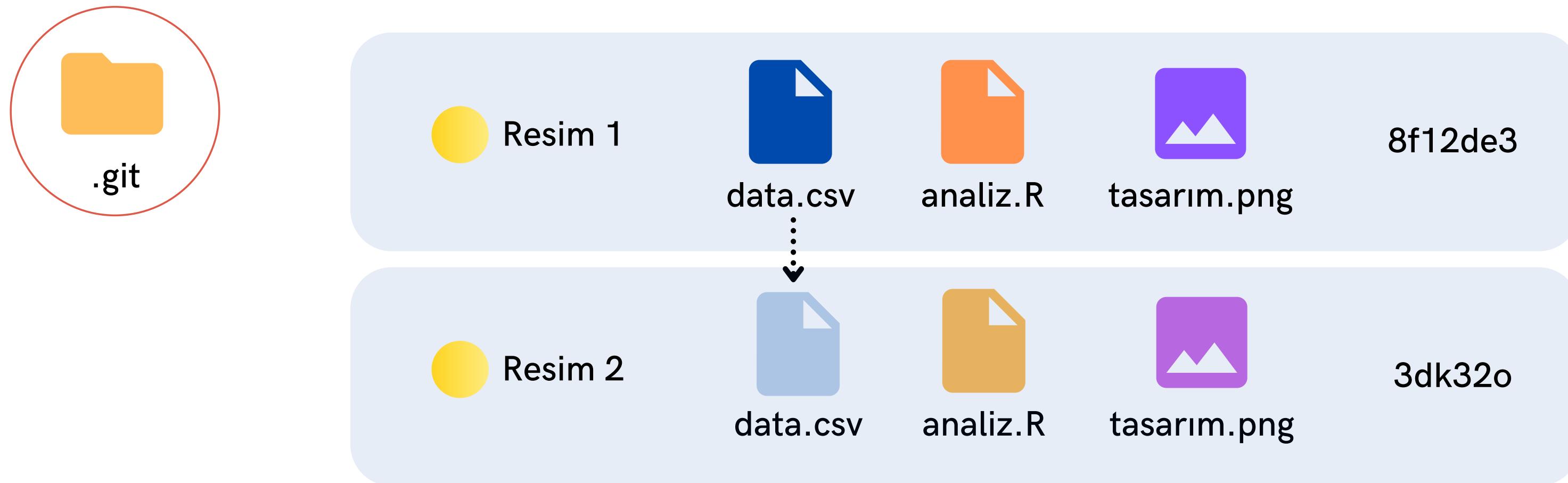
# Git nasıl çalışır?



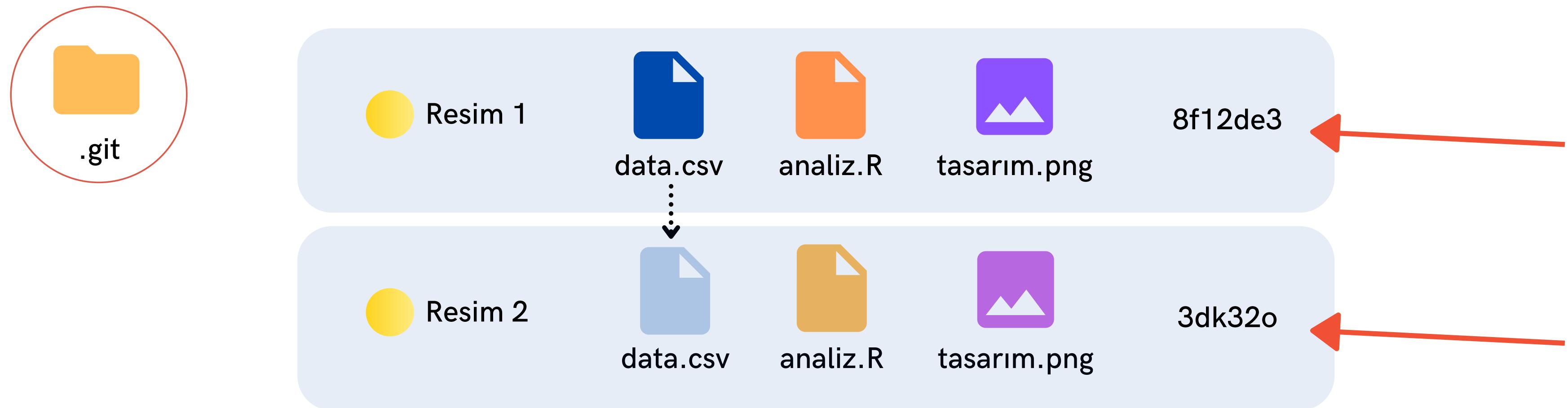
# Git nasıl çalışır?



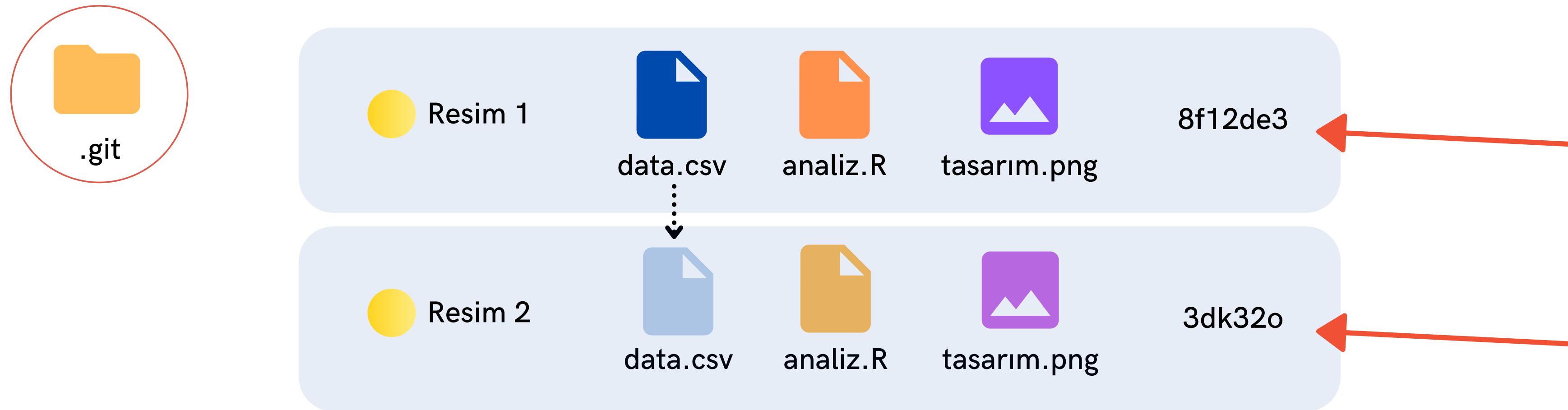
# Git nasıl çalışır?



# Git nasıl çalışır?



# Git nasıl çalışır?



- Git mevcut klasörün durumunun "resmini çeker".
- Değişikliklerle ilgil veriler .git klasöründe taşınabilir olarak saklanır.

# Git jargonu (I)



.git



data.csv



analiz.R



tasarım.png

**repository**  
(ing. mahzen)

- versiyon kontrolü yapılan (bir .git klasörüne sahip) klasör ve versiyon kontrol dası.
- **proje** olarak düşünülebilir.
- kısaca **repo** olarak adlandırılır.

# Git jargonu (I)



.git



data.csv



analiz.R



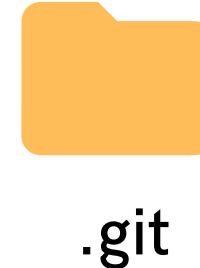
tasarım.png

## stage

(ing. sahne, sahneye koyma)

- reponun içinde yapılan değişiklıkların commit edilecek (resmi çekilecek) olanları.

# Git jargonu (I)



.git

commit 1



data.csv



analiz.R



tasarım.png

8f12de3

## commit

(ing. işlemek/işlem)

- reponun bir andaki durumu veya o durumu kaydeden işlem
- fiil halinde **resim çekmek**, isim halinde **resim** olarak düşünülebilir.

# Git jargonu (I)



.git



data.csv



analiz.R



tasarım.png

**working directory**  
(ing. çalışma klasörü)

- reponun o andaki mevcut dosyaları (stage edilmiş ya da edilmemiş)

# Basit bir git çalışma akışı

1 - Klasörü bir git reposuna çevir



data.csv



analiz.R



tasarım.png

# Basit bir git çalışma akışı

1 - Klasörü bir git reposuna çevir



data.csv



analiz.R



tasarım.png

git init

Mevcut klasörde bir git reposu oluşturur

# Basit bir git çalışma akışı

1 - Klasörü bir git reposuna çevir



.git



data.csv



analiz.R



tasarım.png

git init

Mevcut klasörde bir git reposu oluşturur

# Basit bir git çalışma akışı

2 - Takip edilecek dosyaları ekle (stage)



.git



data.csv



analiz.R



tasarım.png



# Basit bir git çalışma akışı

## 2 - Takip edilecek dosyaları ekle (stage)



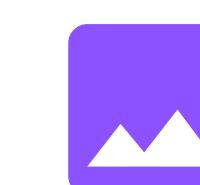
.git



data.csv



analiz.R



tasarım.png



```
git add dosya1 dosya2
```

Verilen dosyaları commit edilmek üzere ekler

# Basit bir git çalışma akışı

## 2 - Takip edilecek dosyaları ekle (stage)



```
git add analiz.R data.csv
```

Verilen dosyaları commit edilmek üzere ekler

# Basit bir git çalışma akışı

3 - Stage edilmiş dosyaları commit et (resmini çek)



```
git commit -m "açıklama"
```

Stage edilmiş dosyaları commit eder

# Basit bir git çalışma akışı

3 - Stage edilmiş dosyaları commit et (resmini çek)



.git



data.csv



analiz.R



tasarım.png



v1

```
git commit -m "açıklama"
```

Stage edilmiş dosyaları commit eder

# Basit bir git çalışma akışı

4 - Aynı işlemleri tekrar et.



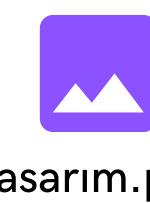
.git



data.csv



analiz.R



tasarıml.png

v1

```
git commit -m "açıklama"
```

Stage edilmiş dosyaları commit eder

# Basit bir git çalışma akışı

4 - Aynı işlemleri tekrar et.



.git



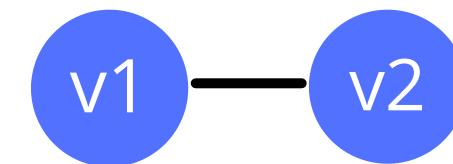
data.csv



analiz.R



tasarrum.png



```
git commit -m "açıklama"
```

Stage edilmiş dosyaları commit eder

# Basit bir git çalışma akışı

4 - Aynı işlemleri tekrar et.

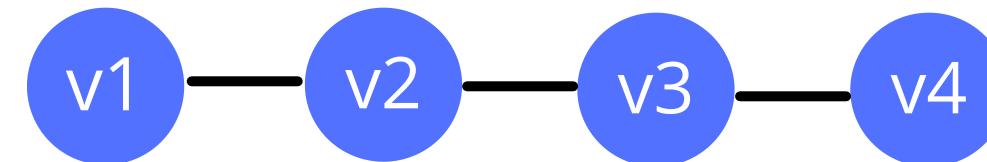


```
git commit -m "açıklama"
```

Stage edilmiş dosyaları commit eder

# Basit bir git çalışma akışı

4 - Aynı işlemleri tekrar et.



git commit -m "açıklama"

Stage edilmiş dosyaları commit eder

# ALIŞTIRMA

# Yapılan değişiklikleri görmek

git status

Reponun durumunu gösterir

git log

Repo üzerindeki değişiklikleri gösterir

git log --oneline

Repo üzerindeki değişiklikleri kısaca gösterir

# Yapılan değişiklikleri görmek

```
git log
```

```
commit 8dc50d57ff06160a7173fe3341248d778052df2e
Author: Ali Seyhun Saral < saral@posteo.de>
Date:   Sun Dec 13 17:28:24 2020 +0100

    remove blur and wording to candidate

    * evaluation/templates/evaluation/GroupVote.html:

commit 4d73fcba249d081cad97eca2f13a30f3e6e392ac
Author: annikahe < [REDACTED]@gmail.com>
Date:   Sun Dec 13 17:15:24 2020 +0100

    add missing comma

commit 5e36c26ba2d0a4c8889546c233f638faaf66d2c5
Author: annikahe < alhennes97@gmail.com>
Date:   Sun Dec 13 17:09:58 2020 +0100

    Control questions: change some wording
.
```

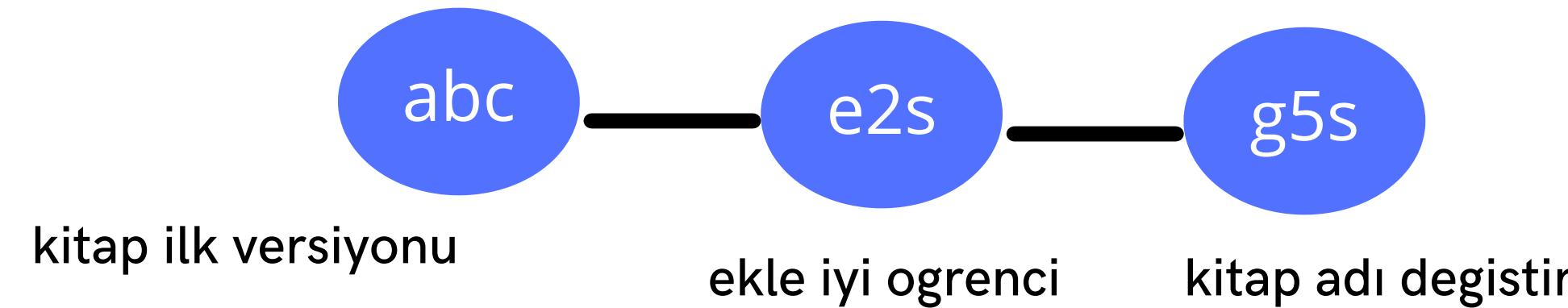
```
git log --oneline
```

```
8dc50d5 remove blur and wording to candidate
4d73fcb add missing comma
5e36c26 Control questions: change some wording
```

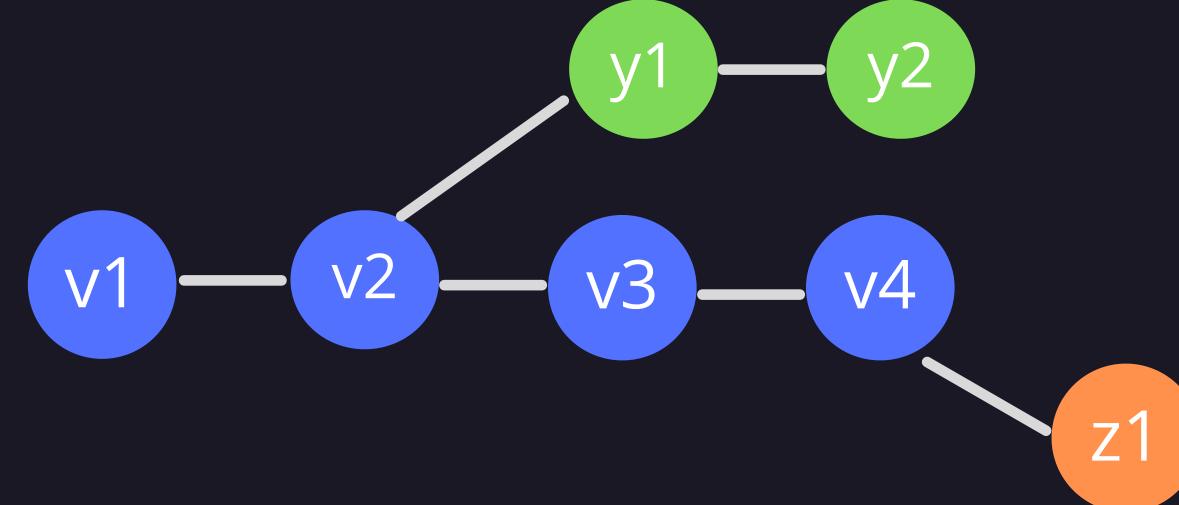
# Önceki bir versiyona gitmek

git checkout **commitkodu**

Commit kodu verilen versiyona geri dönülür.



# Git Dalları: Branch'ler

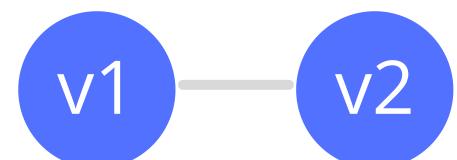


# Branch'lar ve işlevleri

- Projelerdeki yol ayımları
- Birbirinden bağımsız şekilde düzenleme
- Deneysel özellikler
- Farklı kişiler
- Bug ayıklama

# Branch'lar ve işlevleri

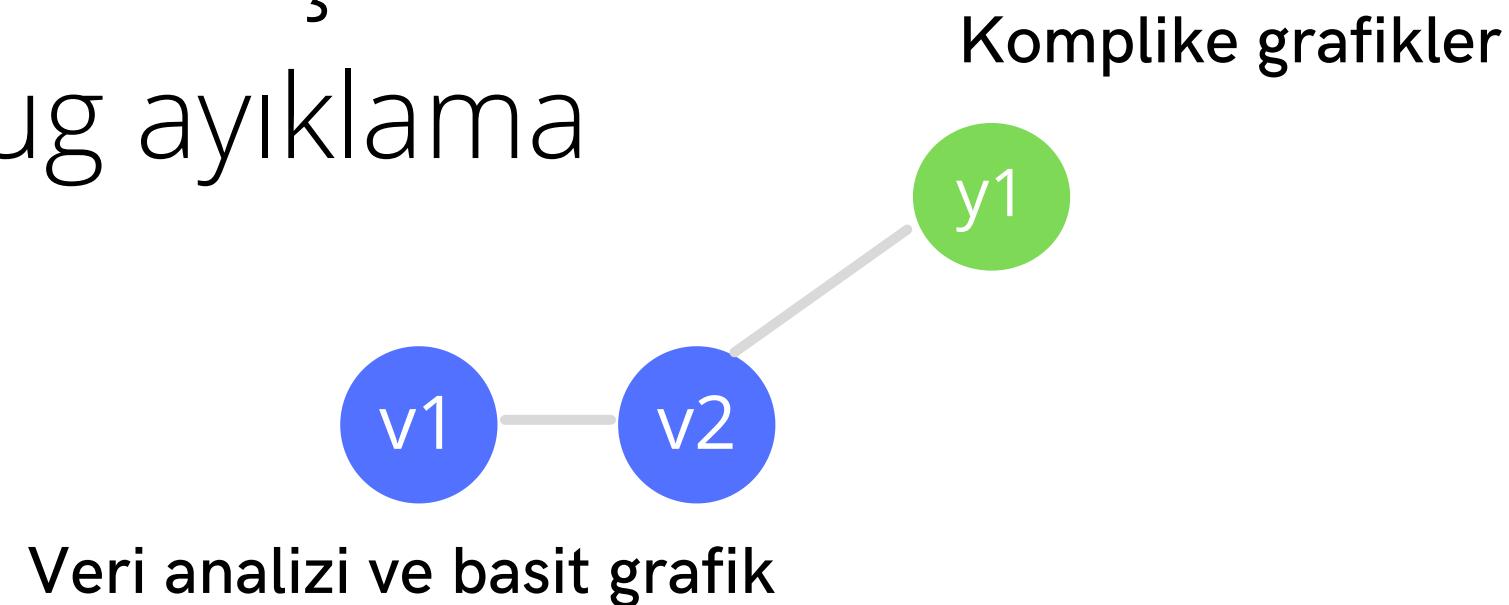
- Projelerdeki yol ayırmaları
- Birbirinden bağımsız şekilde düzenleme
- Deneysel özellikler
- Farklı kişiler
- Bug ayıklama



Veri analizi ve basit grafik

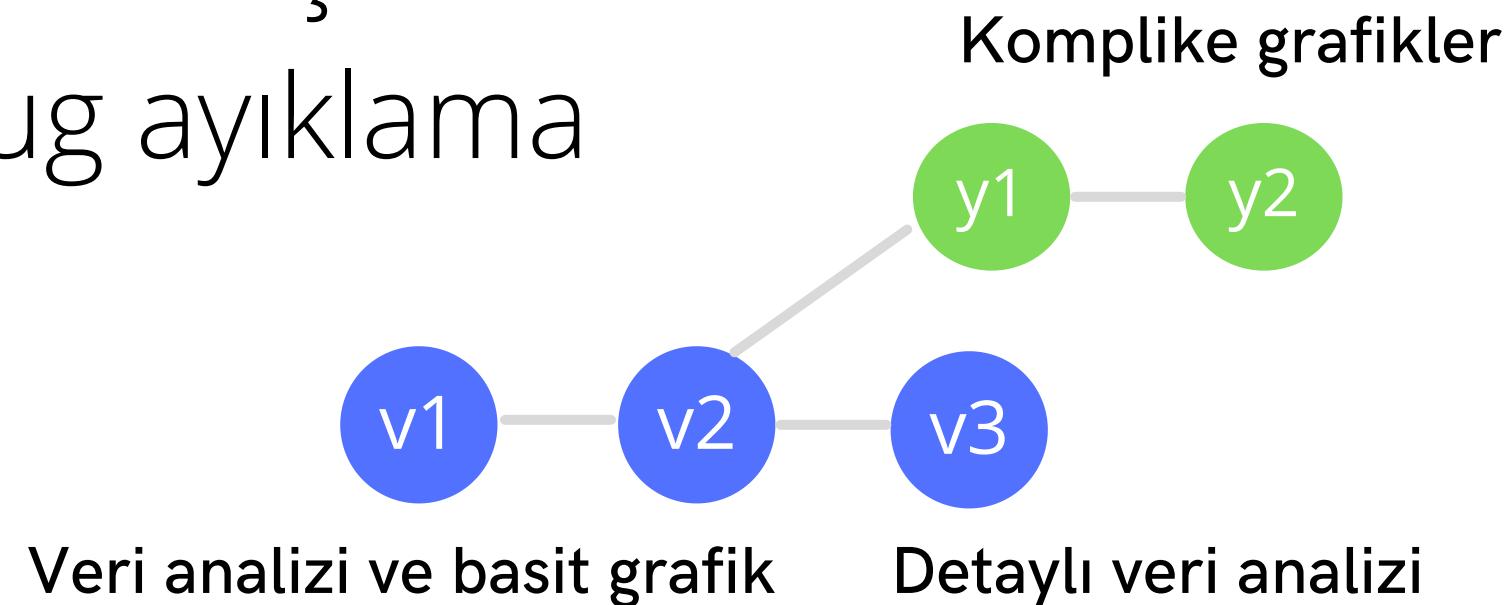
# Branch'lar ve işlevleri

- Projelerdeki yol ayrımları
- Birbirinden bağımsız şekilde düzenleme
- Deneysel özellikler
- Farklı kişiler
- Bug ayıklama



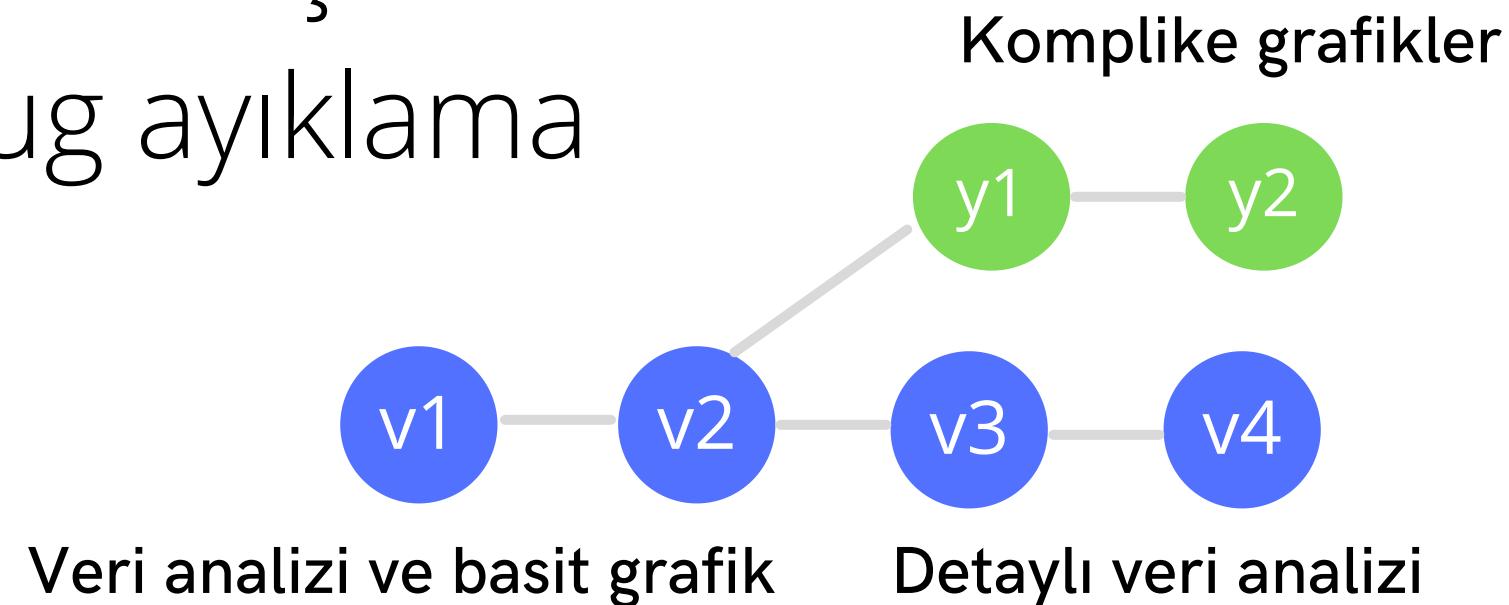
# Branch'lar ve işlevleri

- Projelerdeki yol ayrımları
- Birbirinden bağımsız şekilde düzenleme
- Deneysel özellikler
- Farklı kişiler
- Bug ayıklama



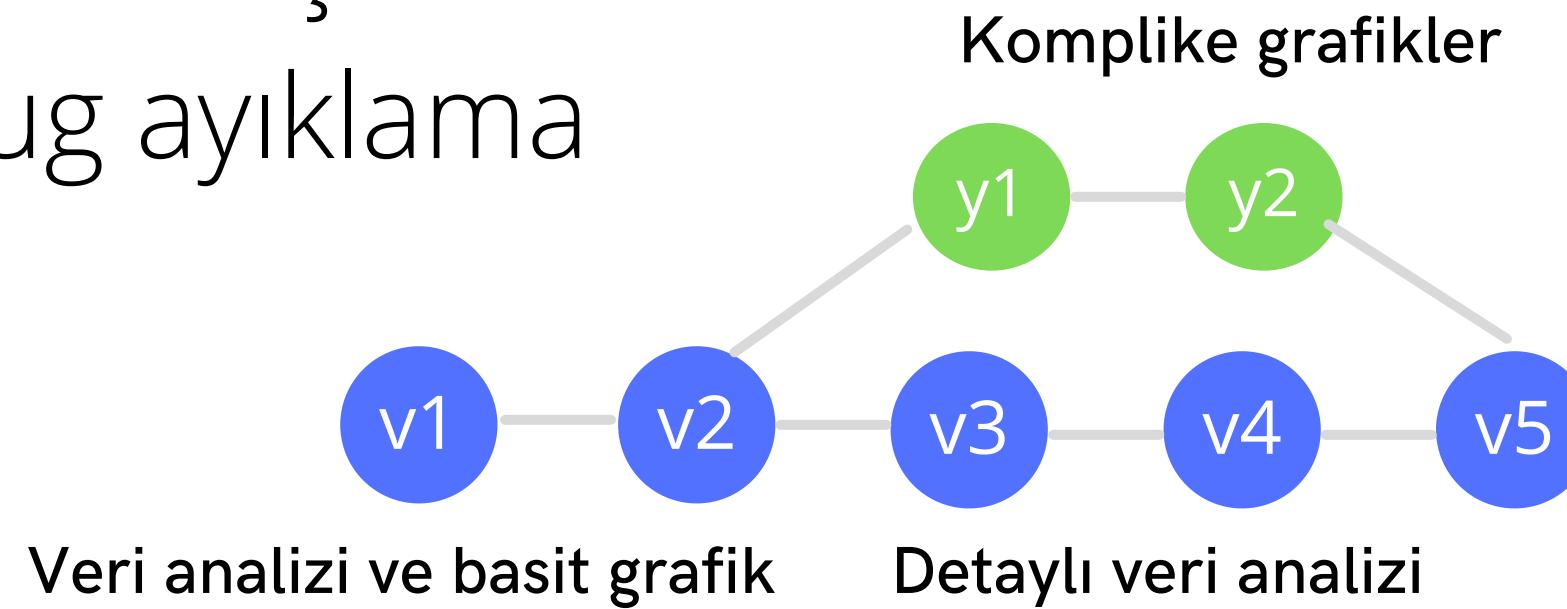
# Branch'lar ve işlevleri

- Projelerdeki yol ayrımları
- Birbirinden bağımsız şekilde düzenleme
- Deneysel özellikler
- Farklı kişiler
- Bug ayıklama

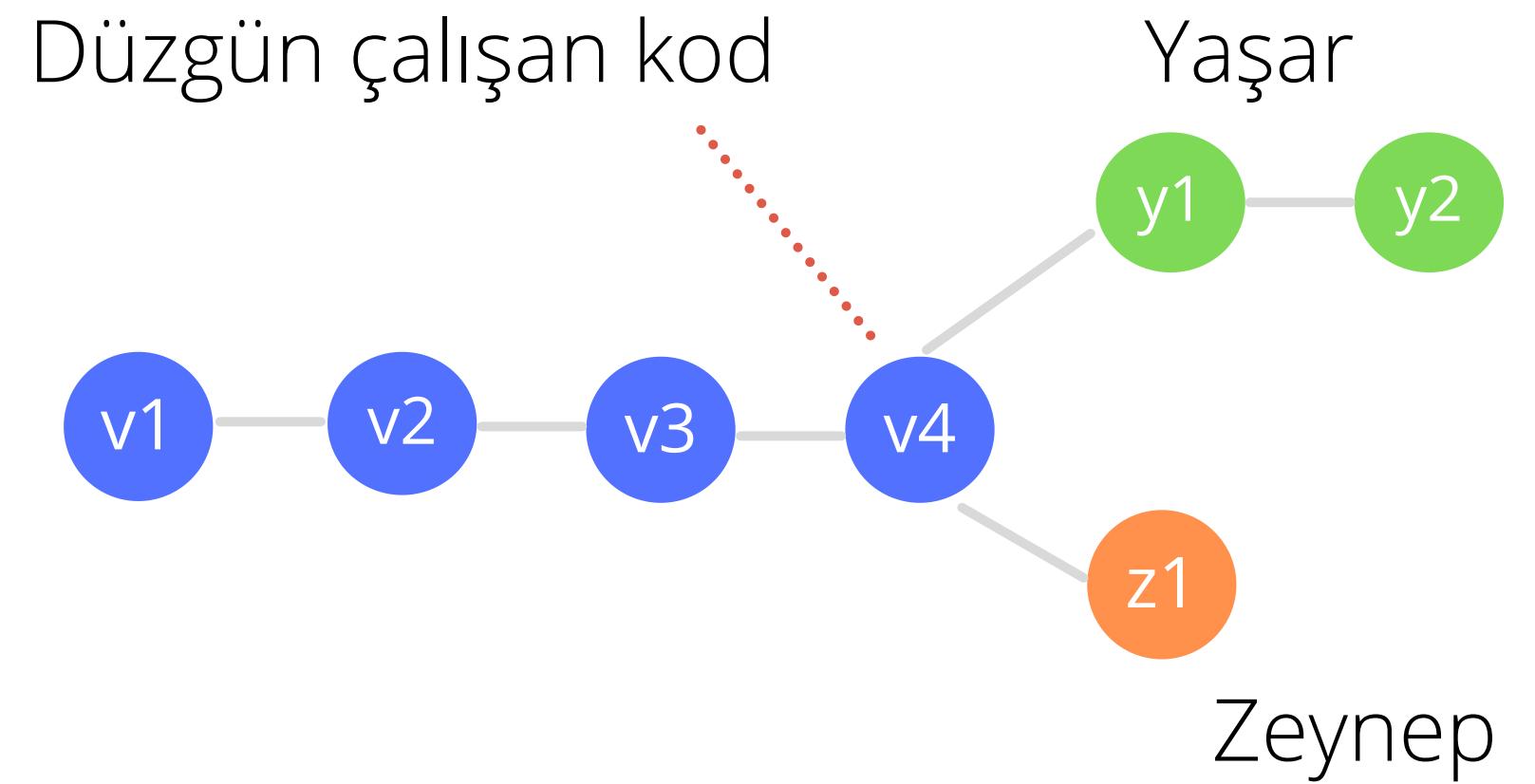


# Branch'lar ve işlevleri

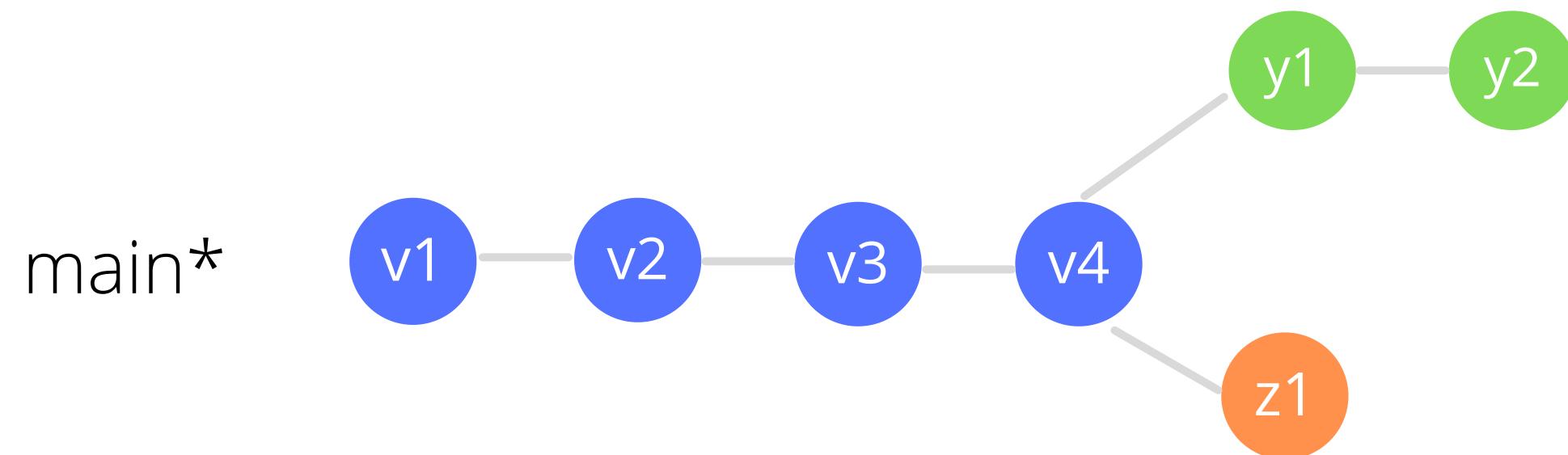
- Projelerdeki yol ayırmaları
- Birbirinden bağımsız şekilde düzenleme
- Deneysel özellikler
- Farklı kişiler
- Bug ayıklama



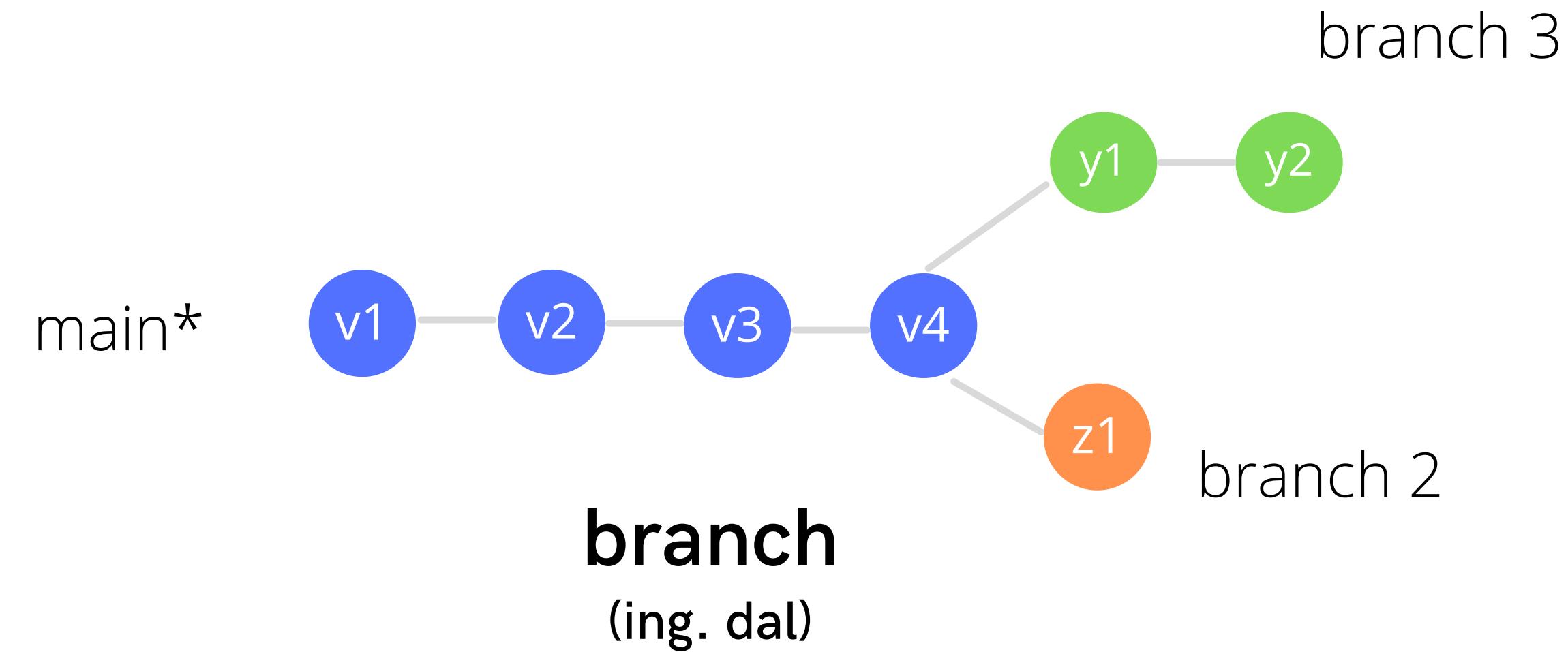
# Branch'lar ve işlevleri



# Branch'lar ve işlevleri

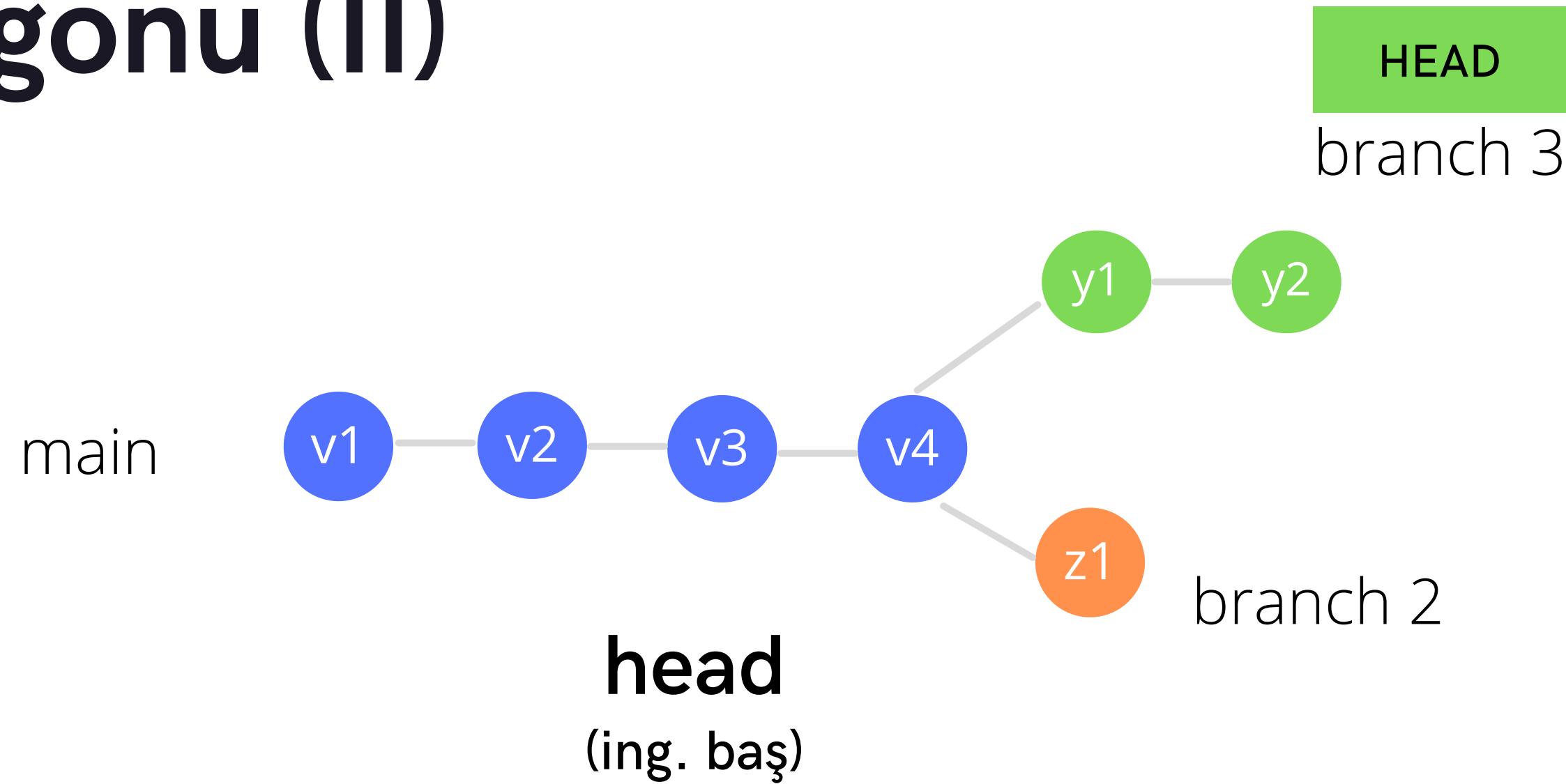


# Git jargonu (II)



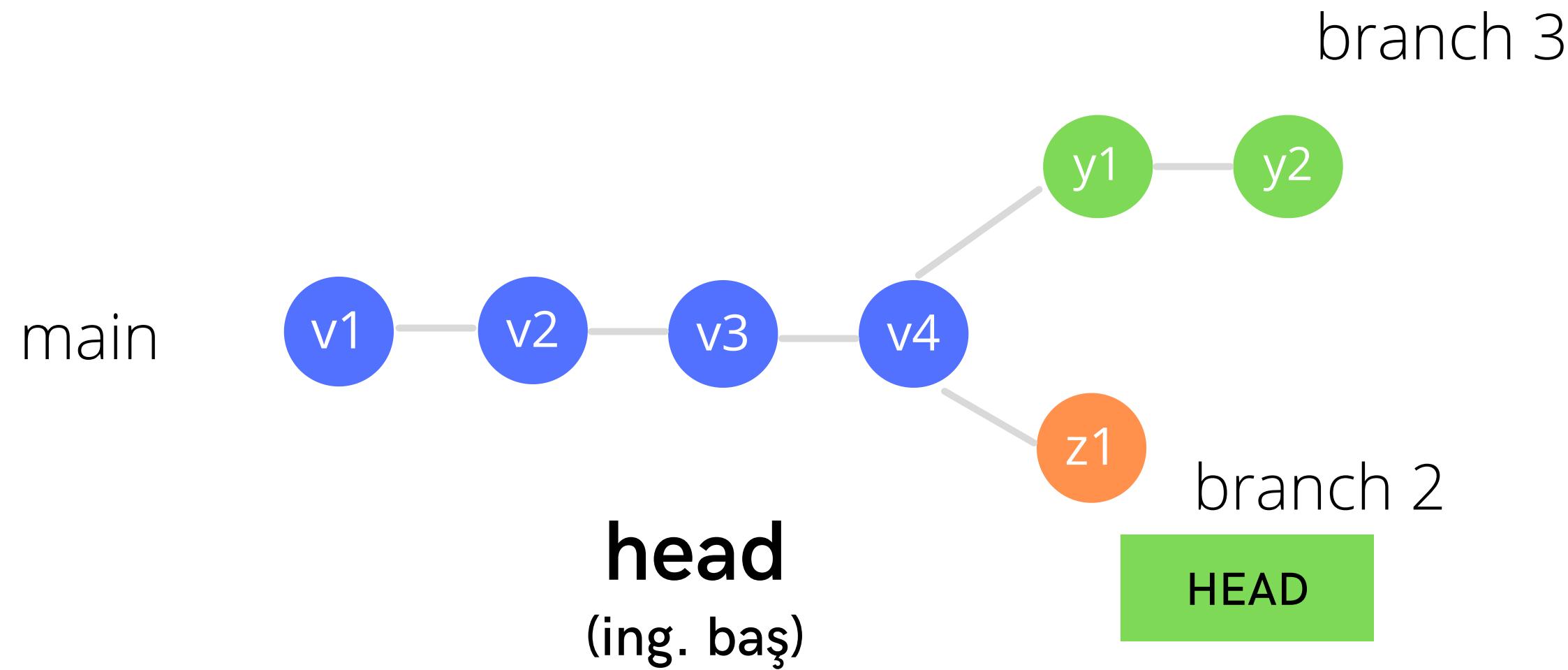
- repo içindeki her ayrık dal

# Git jargonu (II)



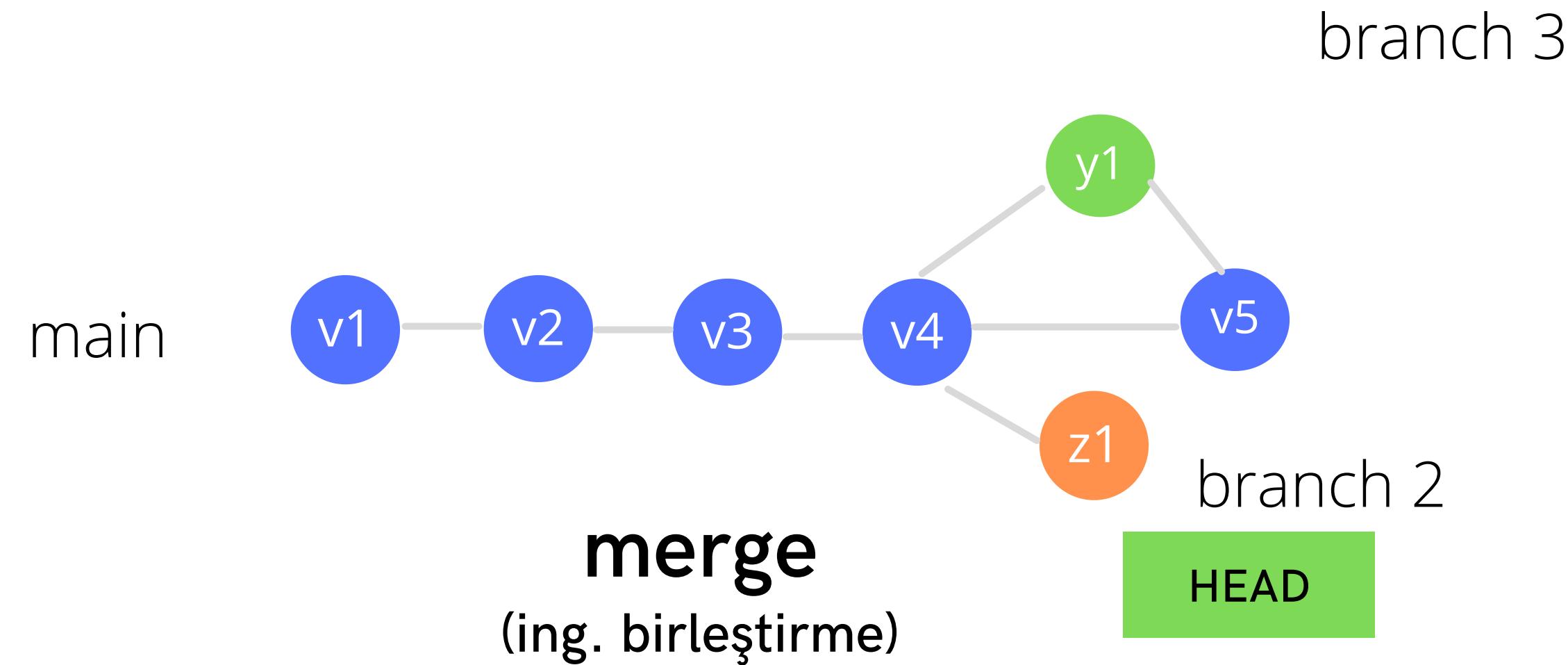
- repo tarihi içinde, şu anda görüntülenen nokta
- head neredeyse "working directory" o konumdaki dosya versiyonlarını görüntüler

# Git jargonu (II)



- repo tarihinde, şu anda görüntülenen nokta
- head neredeyse "working directory" o konumdaki dosya versiyonlarını görüntüler

# Git jargonu (II)



- iki ayrı dalın birleştirilmesi

# Dalların kullanıldığı bir git çalışma akışı

1 - Yeni bir özellik için yeni bir dal oluştur

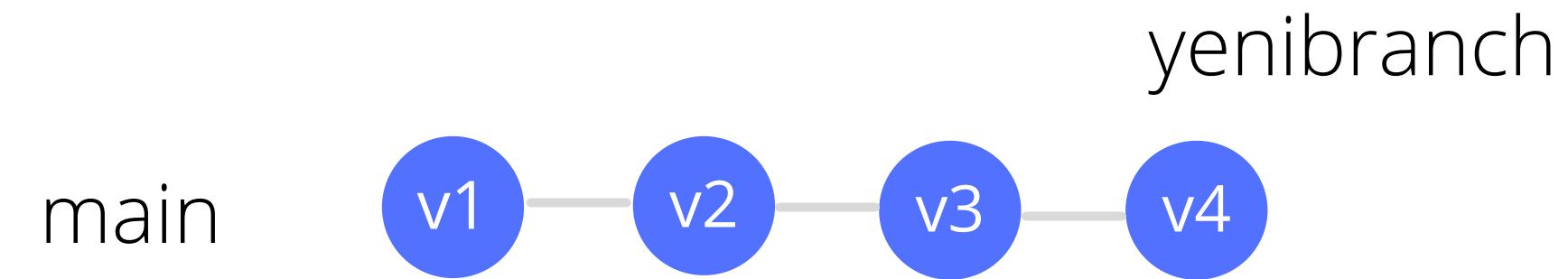


git branch **yenibranch**

Yeni bir dal oluşturur

# Dalların kullanıldığı bir git çalışma akışı

1 - Yeni bir özellik için yeni bir dal oluştur

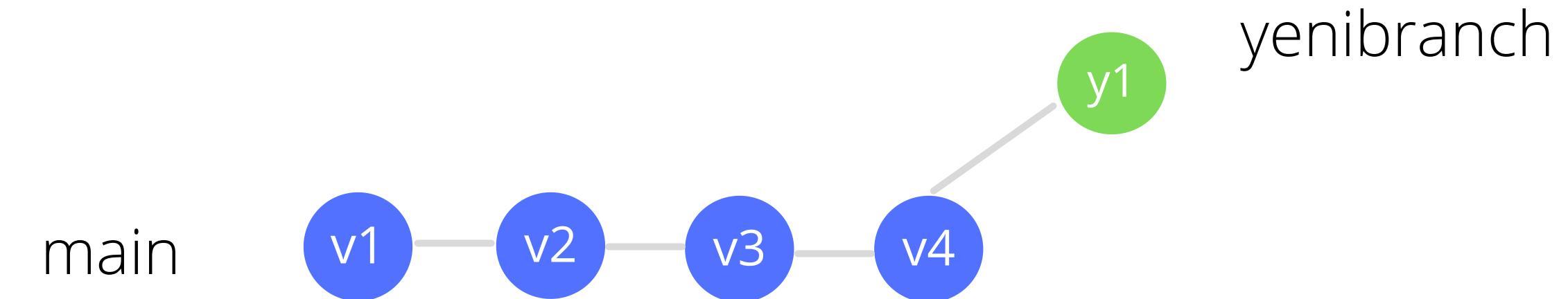


git branch **yenibranch**

Yeni bir dal oluşturur

# Dalların kullanıldığı bir git çalışma akışı

2 - Gereken düzenlemeleri (istendiği kadar commit ile) yap

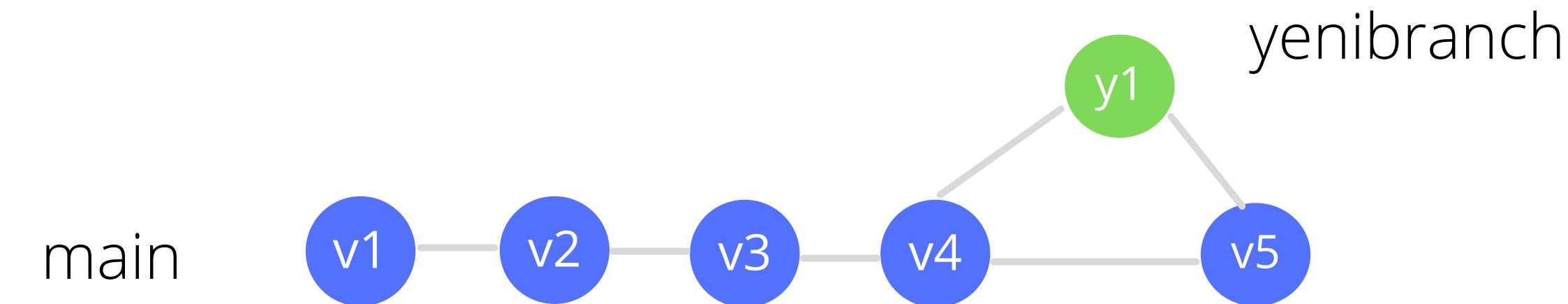


git switch **yenibranch**

Yenibranch adındaki dala geçer

# Dalların kullanıldığı bir git çalışma akışı

2 - Gereken düzenlemeleri (istendiği kadar commit ile) yap

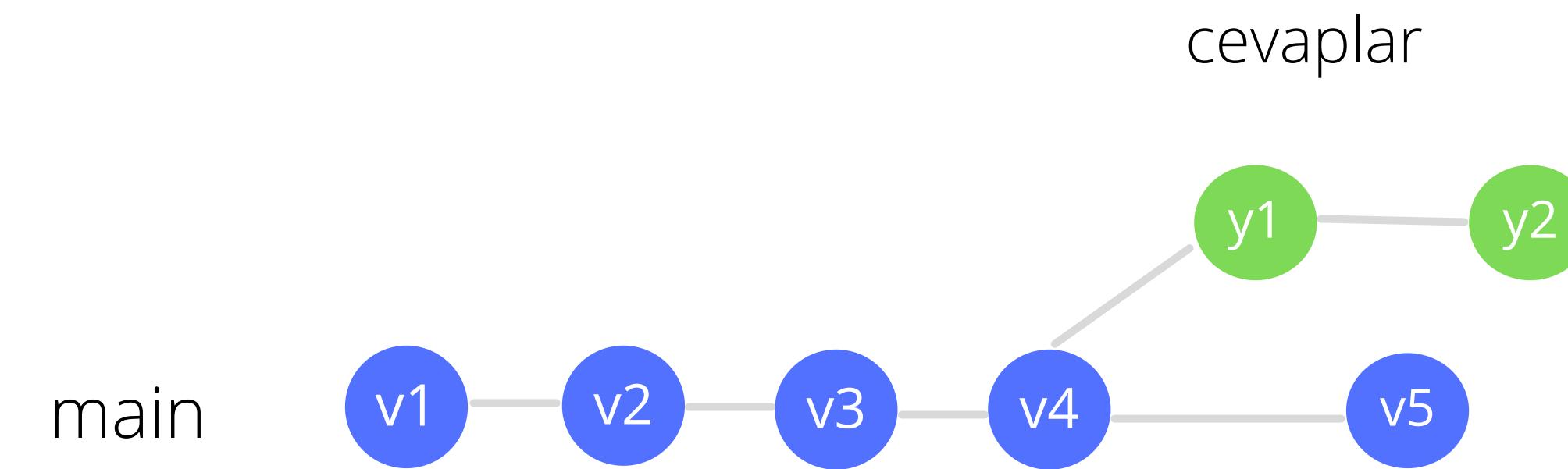


git merge **yenibranch**

yenibranch adındaki dalı mevcut dala kaynaştırır

# ALIŞTIRMA

# Dalların kullanıldığı bir git çalışma akışı



# Git ve GitHub



git



GitHub

- Versiyon kontrol
- Kendi bilgisayarınızda çalışır
- Açık kaynaklı ve ücretsizdir
- Terminal ile
- Git sunucusu
- Ortak çalışmak için ek özellikler
- Müseccel (Microsoft) ancak ücretsiz
- Web tarayıcısı ile

# Github'dan repo indirmek, güncellemek

git clone url

Remote bir repository'i bilgisayara kopyalar

git pull remote branch

Yerel github reposunu günceller

git push remote branch

Yerel github reposunu sunucuya gönderir



sozler.txt



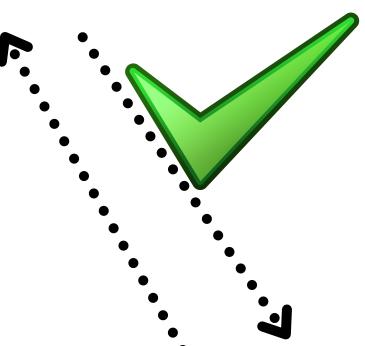
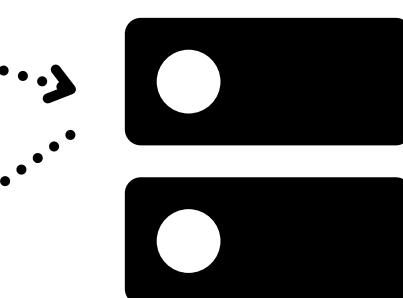
gitar.txt



gitar.txt



sozler.txt



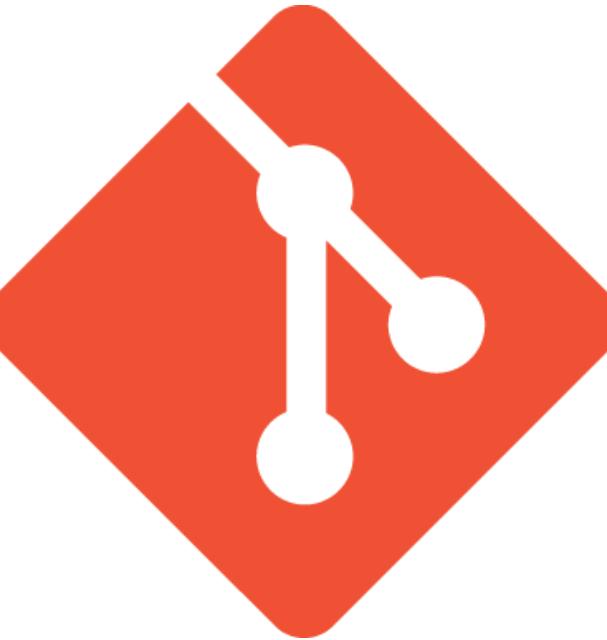
sozler.txt

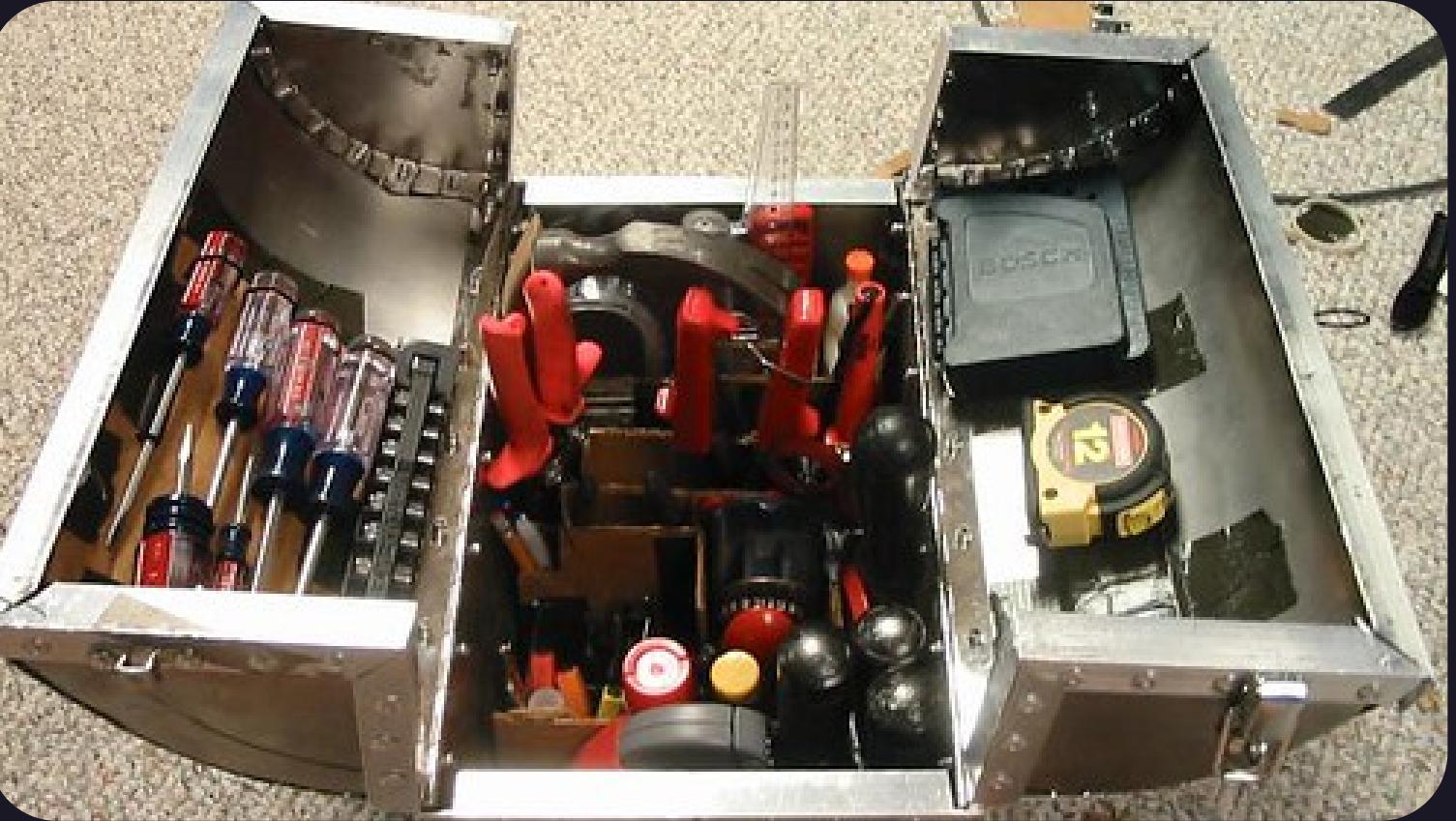
bas.txt



davul.txt

neden  
kullanmalıyım?





1

kısisel fayda



2

# şeffaflık ve tekrar edilebilirlik



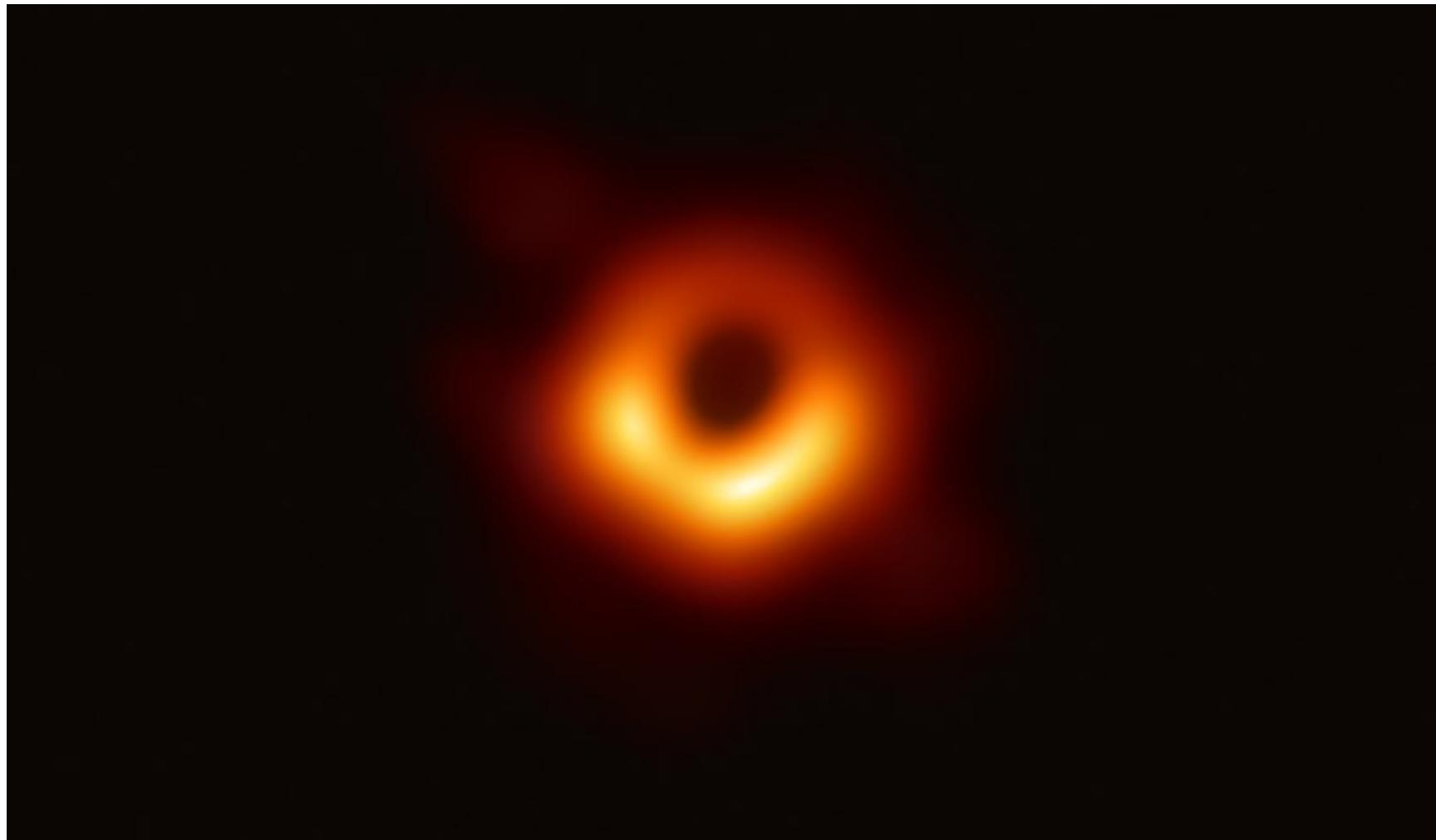
3

## işbirliği imkanları



4

# görünürlük



	achael	undid extreme line wrap	98c8add on 23 Mar	 1,924 commits
	.github/ISSUE_TEMPLATE	Update issue templates		2 years ago
	arrays	added untracked array and example script, probably outdated		12 months ago
	data	overwrite old master		4 years ago
	docs	updated readme and setup		12 months ago
	ehtim	undid extreme line wrap		2 months ago
	examples	fixed nasty bug introduced in pol imaging with imageripython		8 months ago
	models	added rowan and howes		3 years ago
	scripts	merged into master		12 months ago
	.gitignore	modified gitignore		12 months ago
	.mailmap	Add a ".mailmap" file		4 years ago
	Dockerfile	Installing <code>pynfft</code> from conda before installing "requirements.txt"		6 months ago
	LICENSE.txt	Create LICENSE		3 years ago
	README.rst	fixed readme typo		6 months ago
	requirements.txt	update dependencies		2 years ago
	setup.cfg	updated readme and setup		12 months ago
	setup.py	edited README		6 months ago

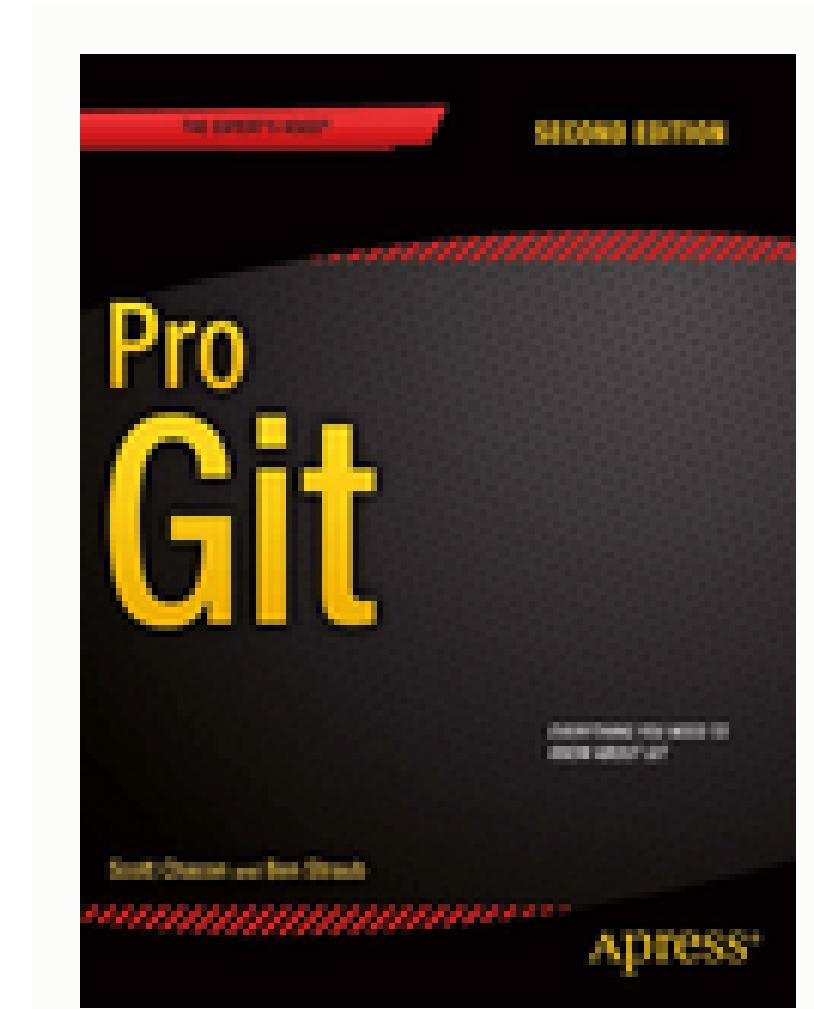
## Contributors 20



+ 9 contributors

# Bazi kaynaklar

- Git dokumentasyonu: <https://git-scm.com/>
- ProGit Kitabı <https://git-scm.com/book/en/v2>
- Git - the simple guide <https://rogerdudler.github.io/git-guide/>
- Software Carpentry <https://software-carpentry.org/lessons/>
- The Turing Way <https://the-turing-way.netlify.app/>





# Git ile versiyon kontrolü

## araştırmacılar için bir giriş

Ali Seyhun Saral  
Max Planck Enstitüsü

