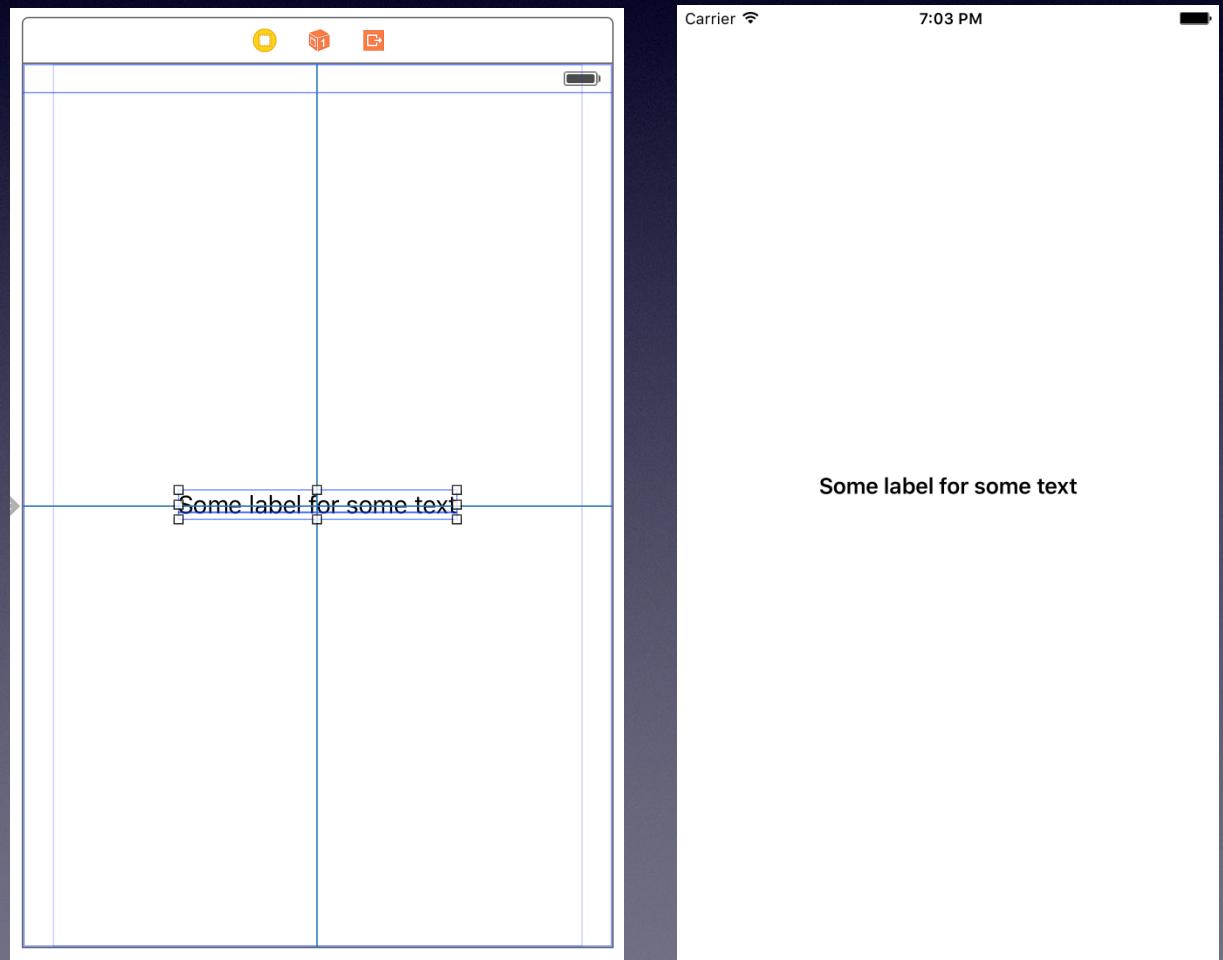


Stackity-Stack

Why UIStackViews are cool

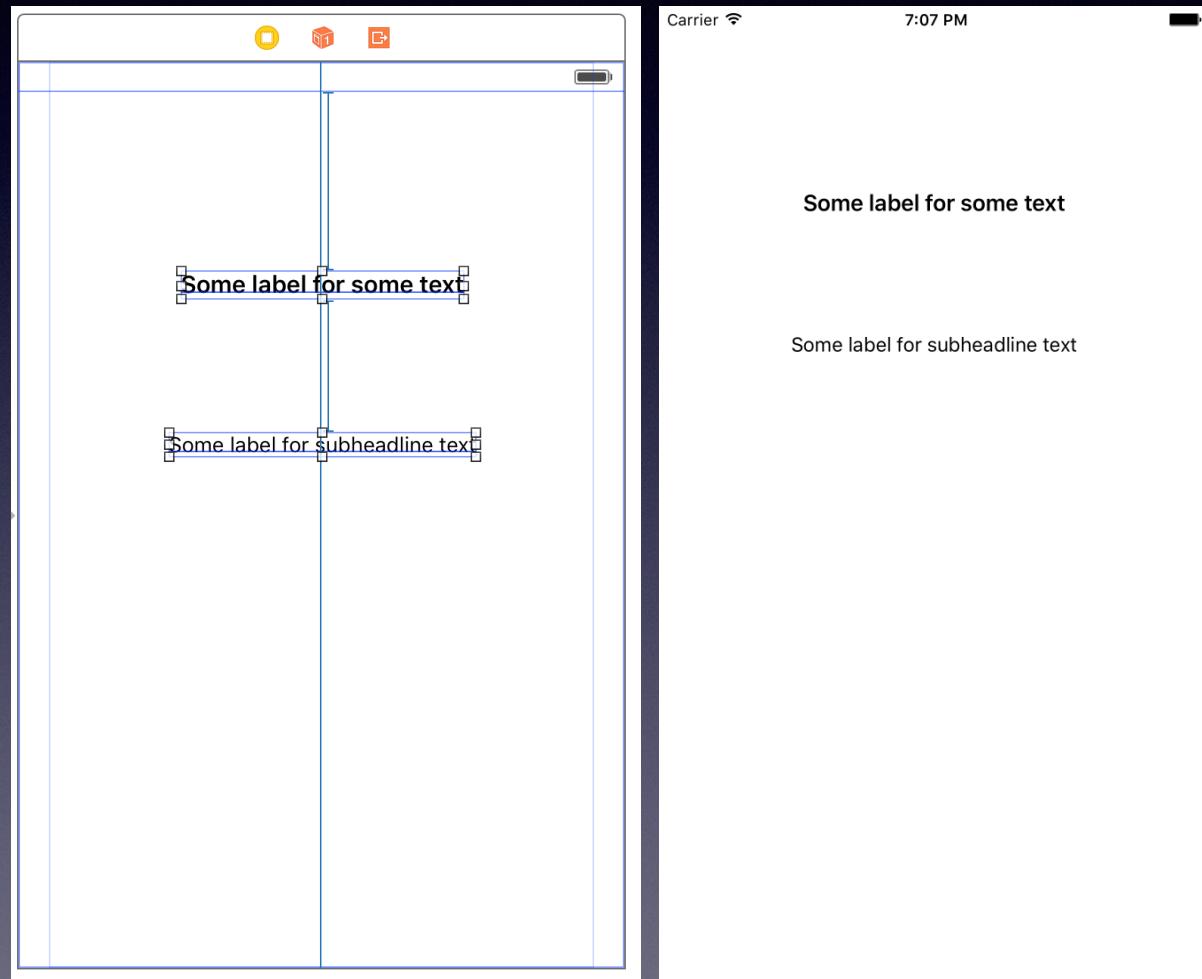
The initial ask

- One label on the view.
Easy enough, center it horizontally with lines = 0



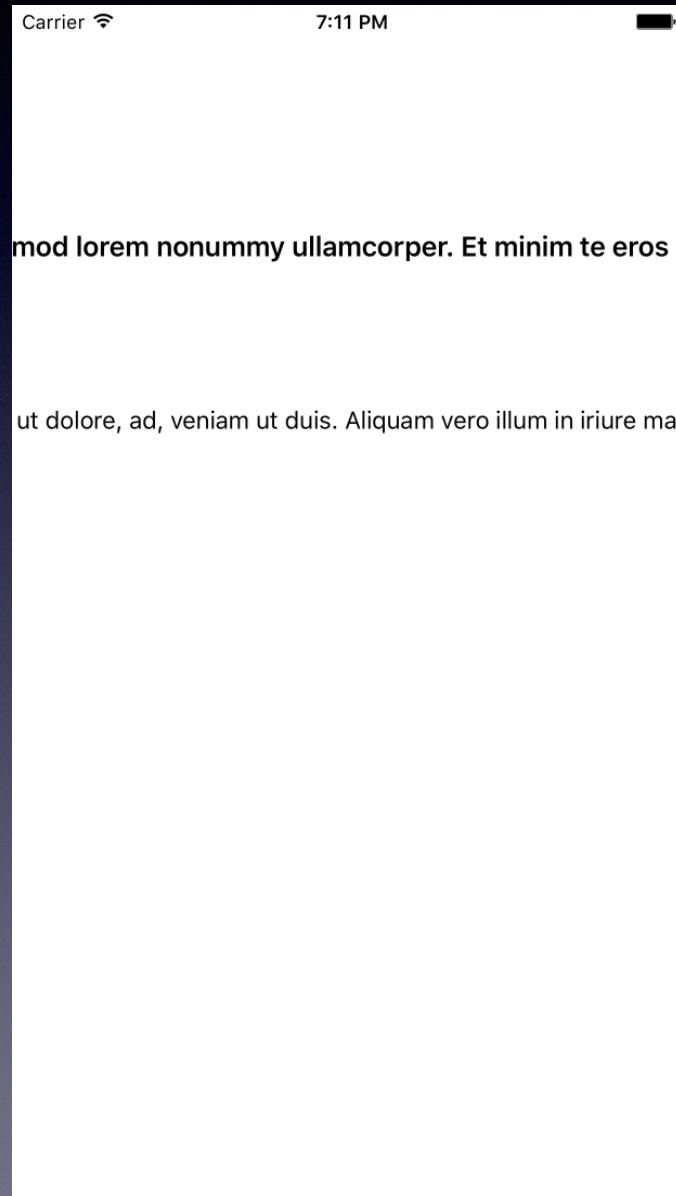
Second ask - another label

- Simple enough.
Remove the
centering Auto
Layout
constraints on
the first label
(vertical), etc.



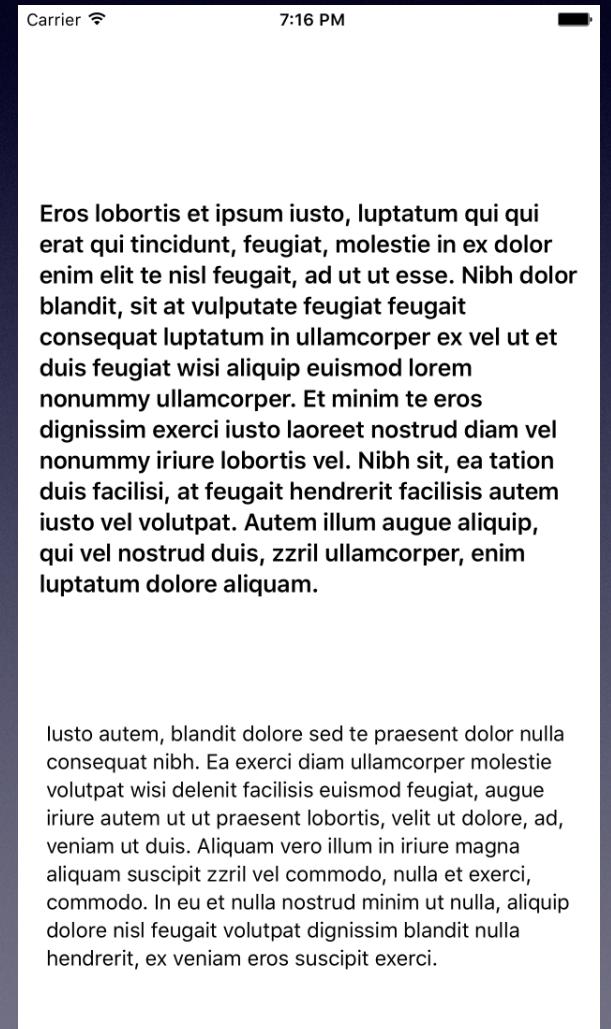
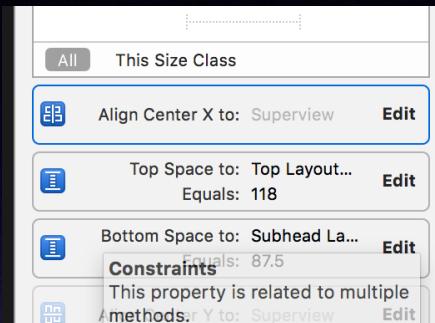
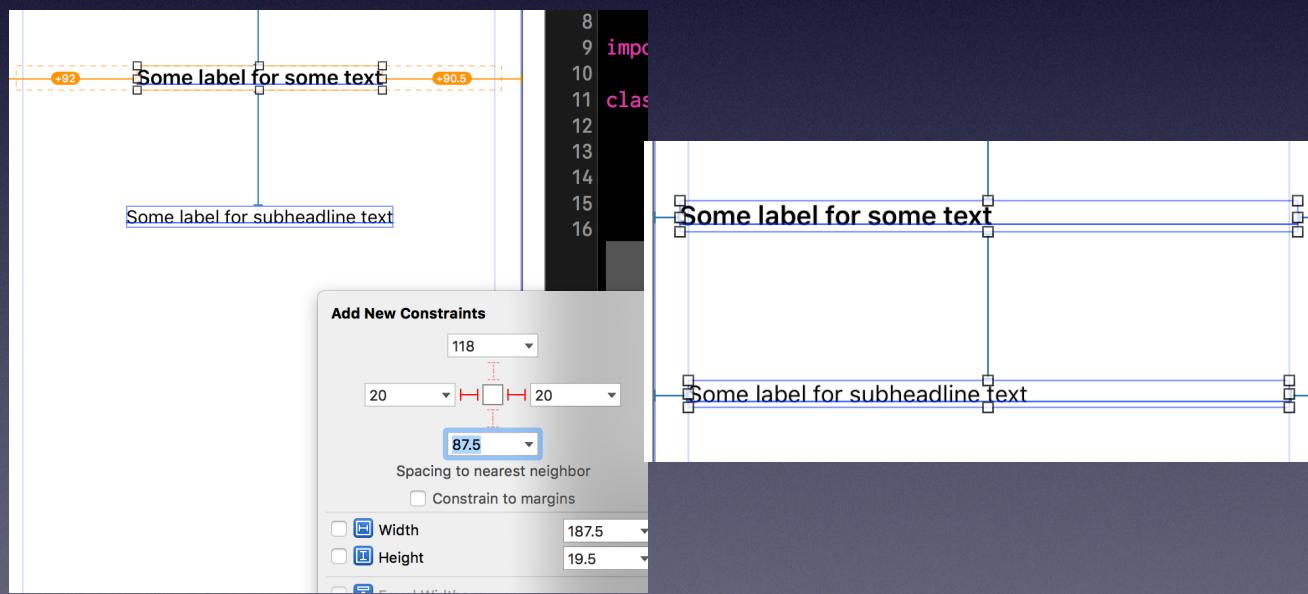
But the text changes

Eros lobortis et ipsum iusto, luptatum qui
qui erat qui tincidunt, feugiat, molestie in
ex dolor enim elit te nisl feugait, ad ut ut
esse. Nibh dolor blandit, sit at vulputate
feugiat feugait consequat luptatum in
ullamcorper ex vel ut et duis feugiat wisi
aliquip euismod lorem nonummy ullamcorper.....

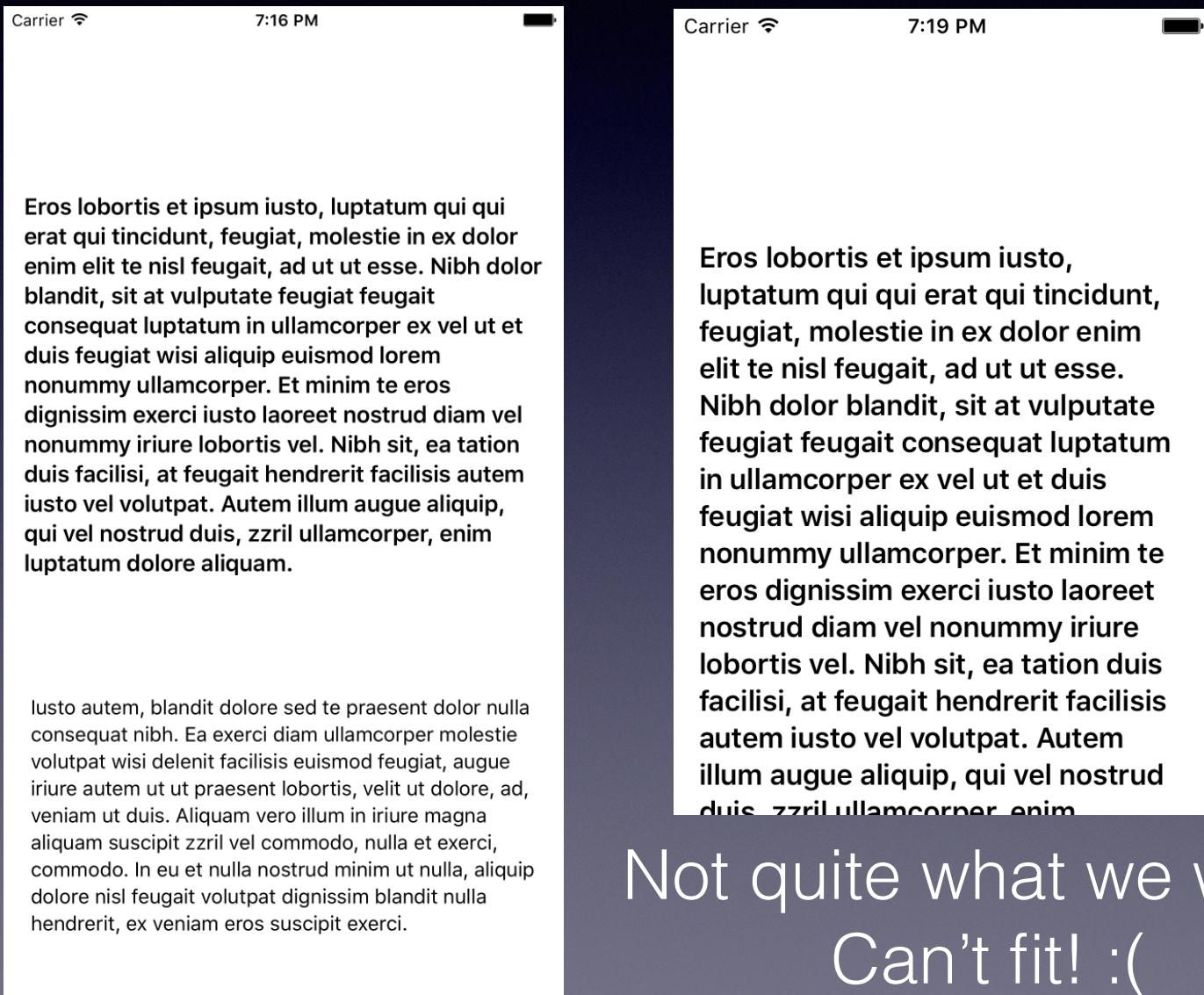


Constraints to edges

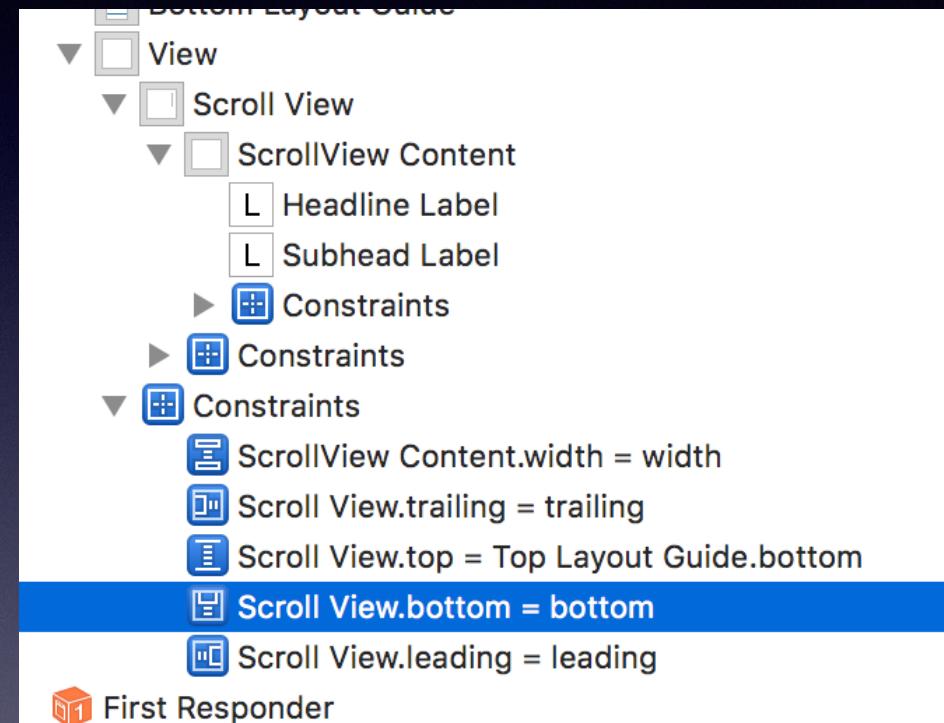
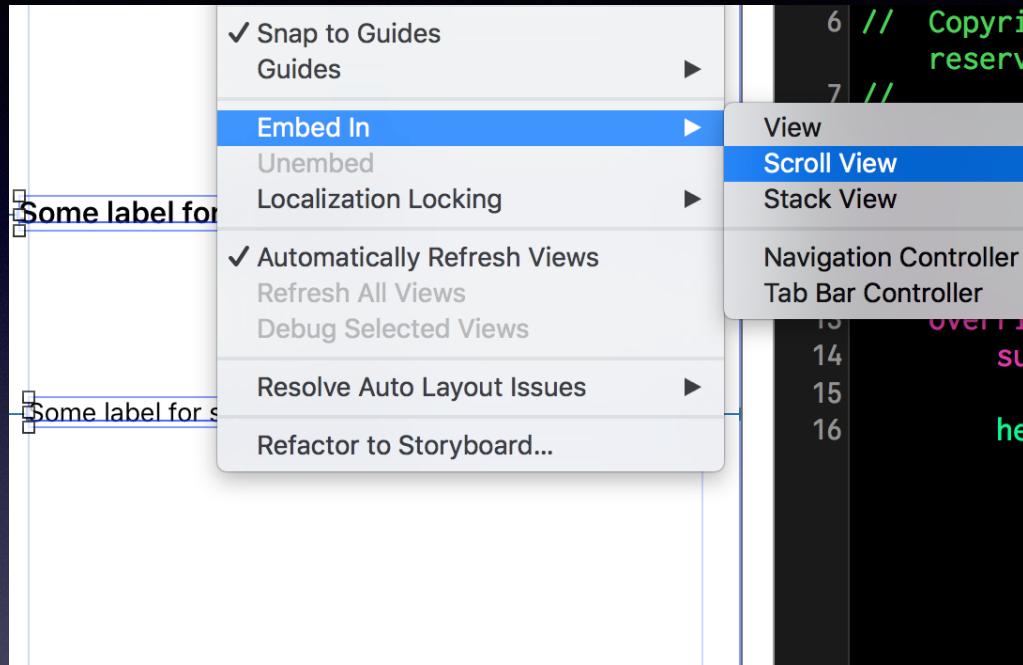
- Remove the center constraint



- That's closer, but how's it look on a smaller screen?



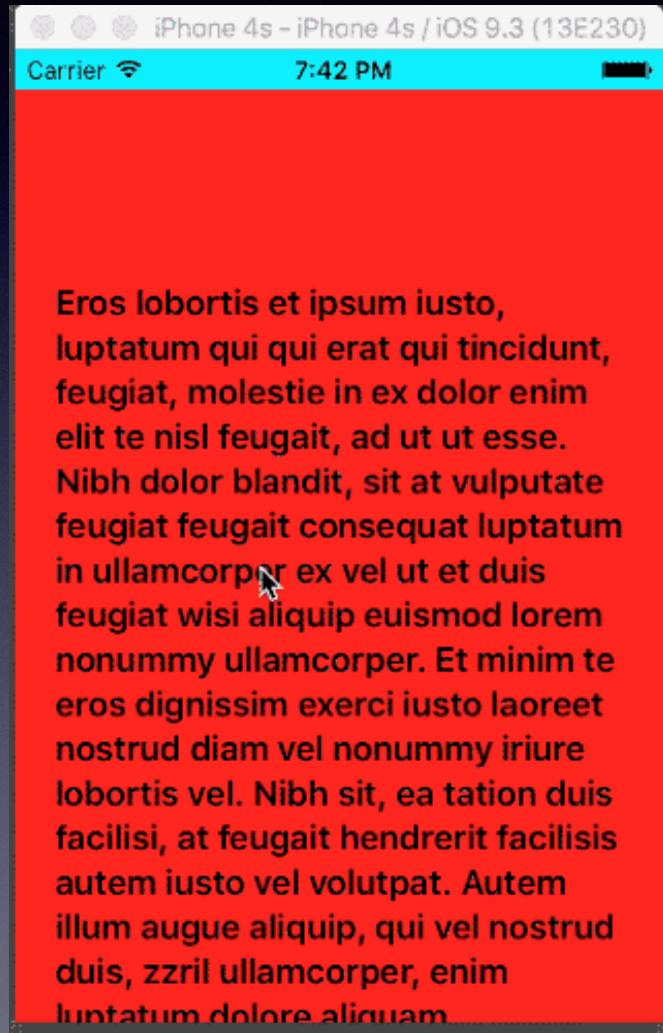
Embed them in a scrollview



- First embed them in a View
- Then embed that in a ScrollView
- • Add a UIView as the UIScrollView content view, aka “ScrollViewContentView”
- • Pin the “ScrollViewContentView” to the edges of the UIScrollView
- • Add a equal width constraint from the ScrollViewContentView to the UIScrollView’s superview.

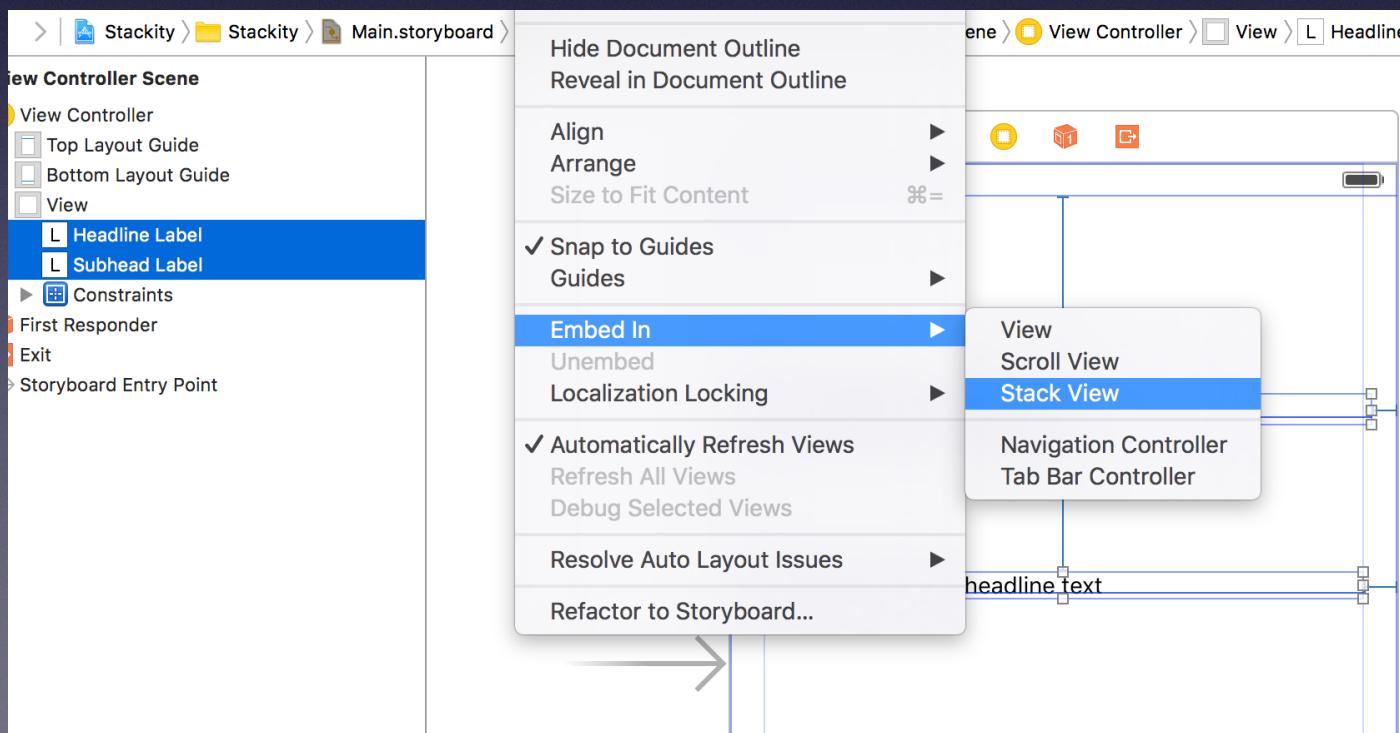
It works!?

- Yes, it works. But it's not ideal.
- Creating this layout in Interface builder can be - tricky.
- You can go crazy with the labels' intrinsic content size
- You can be even more adventurous with Content Hugging & Compression Resistance Priorities.



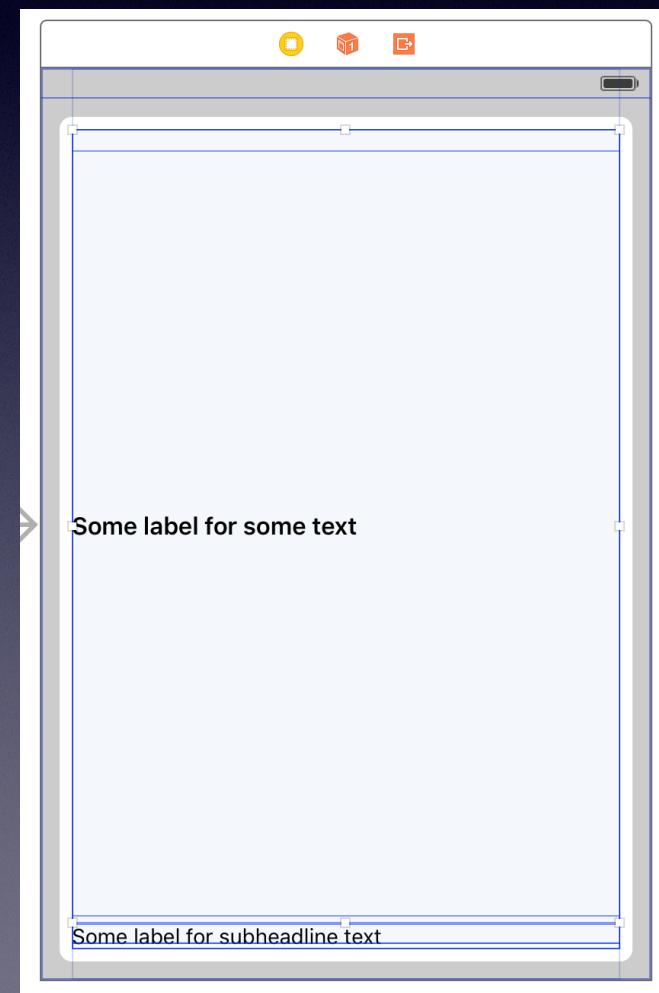
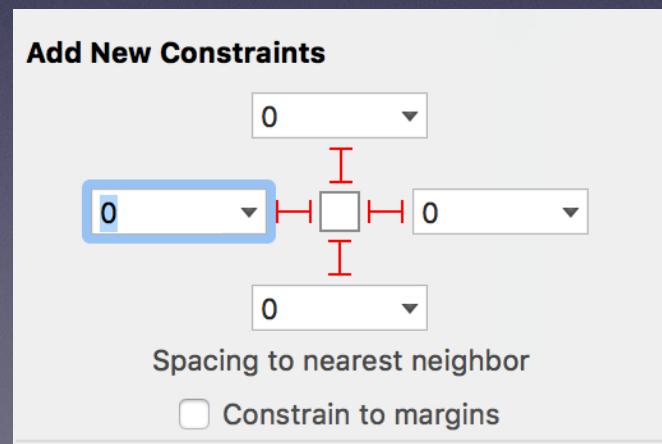
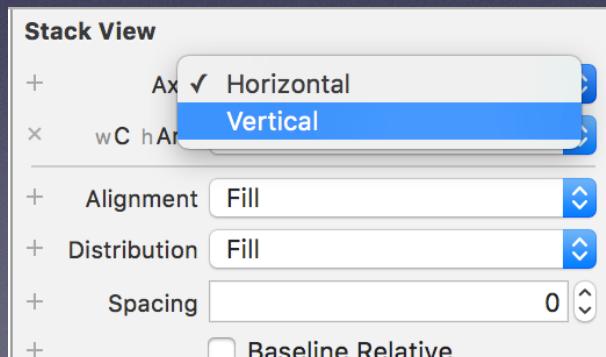
Use a StackView

- Apple introduced the UIStackView at WWDC 2015. It is iOS9 Only.
- View layout & presentation (fitting everything specs dictate) is one of the more difficult tasks for iOS Developers.
- StackViews can make laying out dynamic content easy. It can sometimes eliminate the pain of inner constraints by handling the creation of the the constraints at runtime based on the Stack's properties you've set in Interface Builder.



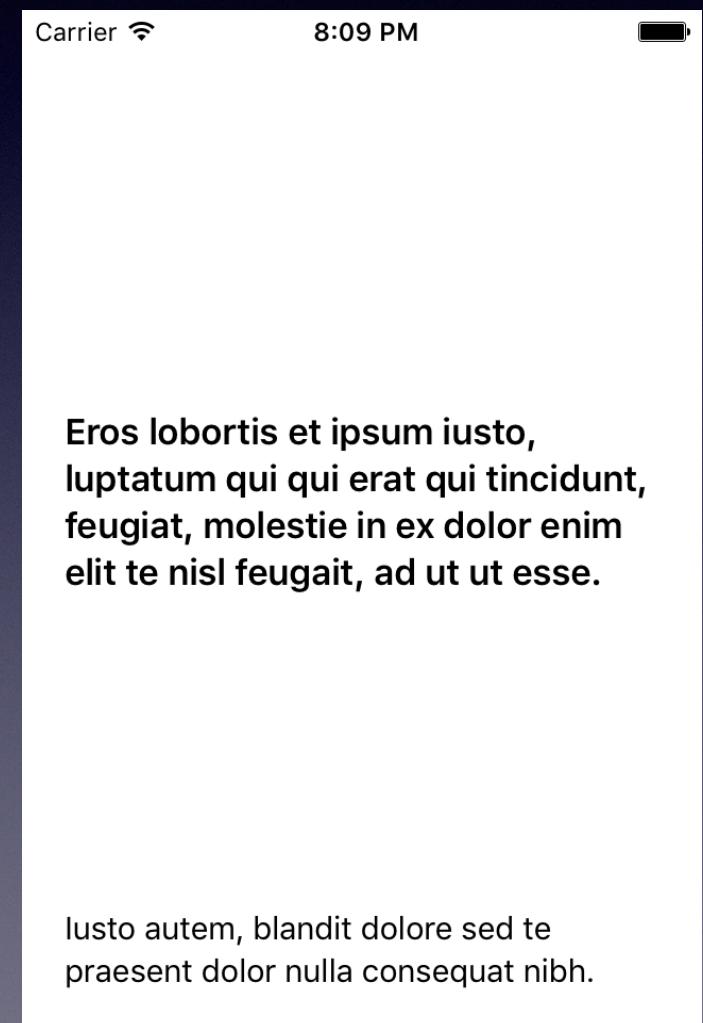
Use a StackView

- Change the Stack to Vertical layout
- Pin the stack to it's superview
- Change the content hugging on the labels



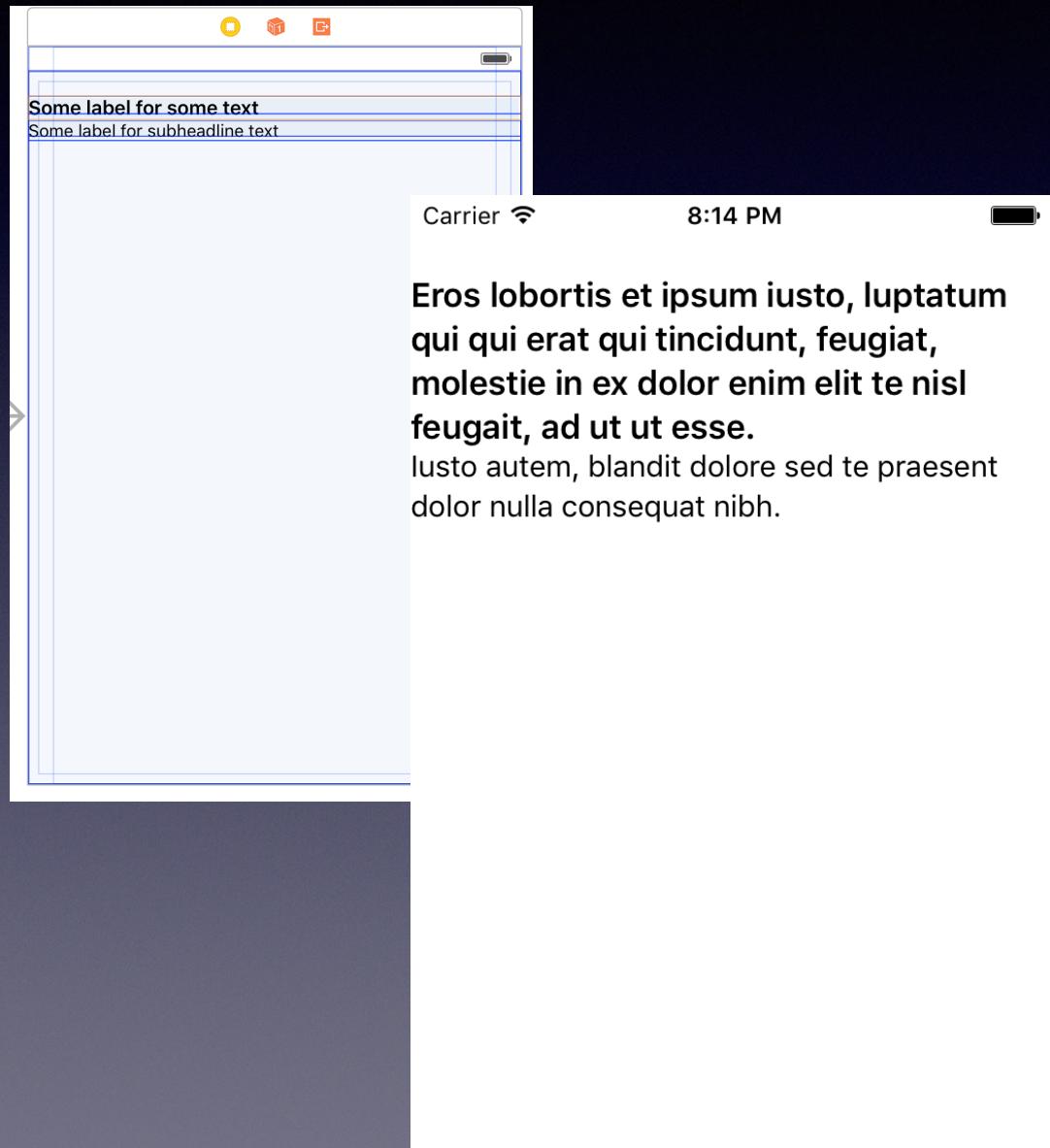
Use a StackView

- Change the Stack to Vertical layout
- Pin the stack to it's superview
- Change the content hugging on the labels
- This is close, but again, on an iPhone 4 Screen, it won't fit.



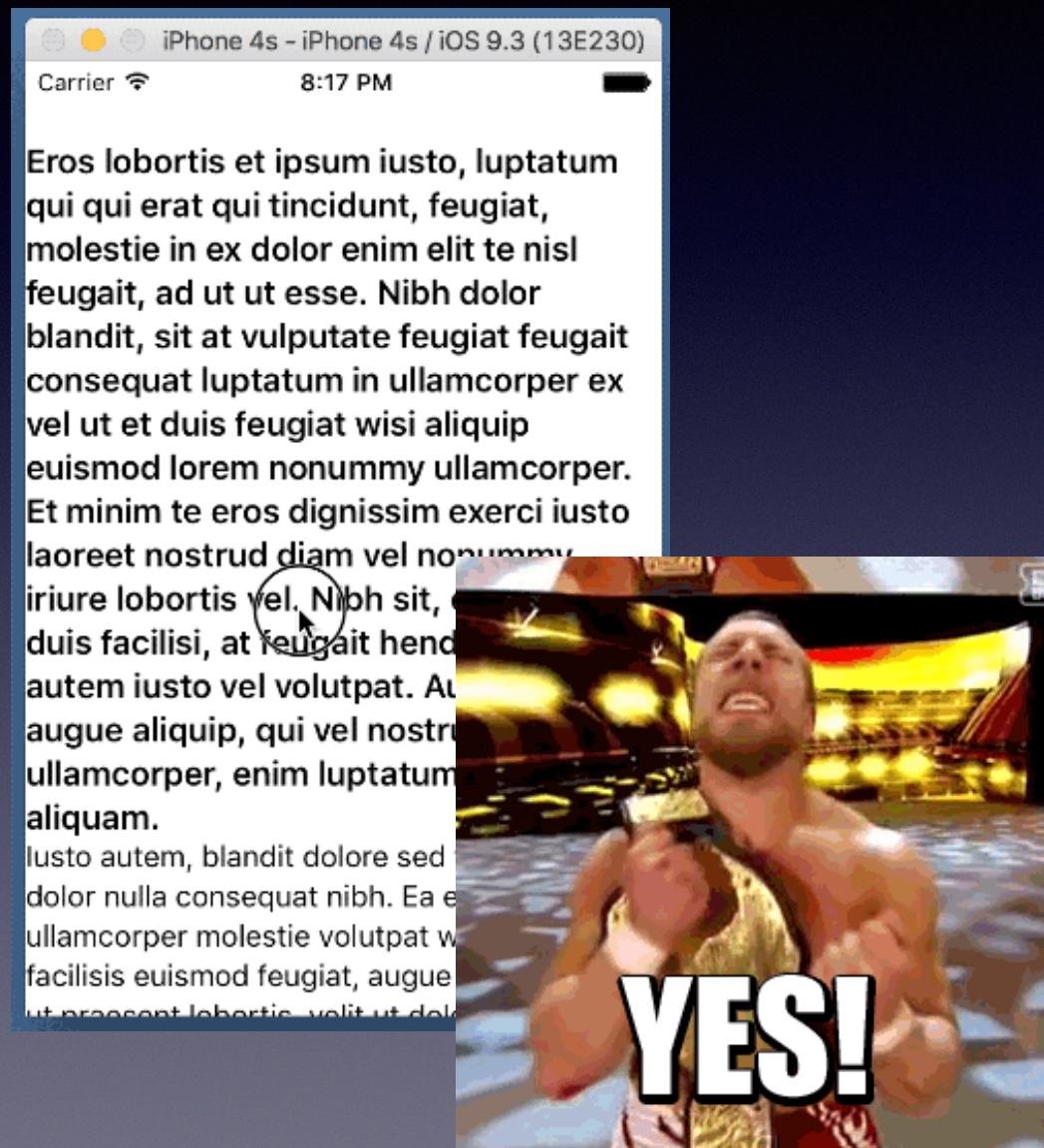
Embed the Stack in a Scroll

- Same process as embedding a `UIView` in a `ScrollView`
- Run it (with less label text) and get... nothing special:



But with larger text?

- With longer text in the labels, it causes the ScrollView's content size to balloon, hence scrollable
- This is a really easy way to have lots of content on the screen, that's instantly scrollable.



Further tidbits

- Stack views can be by vertical or horizontal axis.
- Item spacing can be equal, proportional or “just fill it”
- Alignments can be baseline, center, bottom, etc.
- No limit to what you can put in them.
- StackViews can be stacked within other StackViews.

Resources

- WWDC 2015: Mysteries of AutoLayout Part1
<https://developer.apple.com/videos/play/wwdc2015/218/>
- <https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/AutolayoutPG/LayoutUsingStackViews.html>
- Tutorial: <https://www.raywenderlich.com/114552/uistackview-tutorial-introducing-stack-views>

#endif

My name is

Shane Zatezalo.



I help people figure out what they want, then I make their app for them.

I run the @BuckeyeCocoa user group. I own Lottadot LLC. And I create sweet apps with my team at @Dynamit in Columbus, OH. And we're hiring. Hit me up.

@lottadot



D