

web常见问题排查

2014-09-26 基础架构快报

奉上下午分享会的PPT，后续我们会持续讨论不同的案例，也希望业务团队们多给我们提供分享的机会：)

web常见问题排查

Nginx & php

 基础架构快报

一般排查方法

- 检查error_log
- 检查access_log
- strace 查看系统调用
- tcpdump 分析网络状况

根据不同问题，要选择合适的方法，最简单的方法还是查看error_log，有些怪异的问题，从error_log中能找到蛛丝马迹。慢请求相关的问题，可通过access_log记录请求时间。如果log中没有什么有用信息，可以尝试分析下系统调用，或网络。程序本身的问题，可能需要gdb调试。

 基础架构快报

error_log 排查

error_log 提供了异常丰富的信息，比如nginx处理请求出错，网络连接出错，后端返回出错，系统调用出错等各种错误信息，如果是线下调试，则可以开启debug信息，便于调试排查

- connect,write,read timeout 等网络超时错误
- Permission denied , File not found等系统调用错误
- HTTP 状态码400,499,500等对应的错误

 基础架构快报

error_log 排查

addops 反应了一个很怪异的问题，更新nginx后，http请求不能返回完整的数据，返回部分数据后请求就结束了，每次都能复现。

- 查看error_log

```
2014/09/24 10:28:14 [crit] 12930#0: *1 open() "/data/nginx/proxy_temp/1/00/0000000001" failed (13: Permission denied) while reading upstream, client: 127.0.0.1, server: localhost, request: "GET /normal/rand10m.html HTTP/1.1", upstream: "http://127.0.0.1:8805//rand10m.html", host: "localhost:8804"
```

 基础架构快报

access_log 排查

access_log 提供了客户端访问日志，nginx 的access_log可灵活定义日志格式，日志内容，比如后端响应时间，后端返回状态码，后端的failover过程等，另外，也可根据access_log 进行统计，比如根据接口统计，根据客户端IP统计，根据服务端IP统计，根据后端响应时间统计等。

- 统计HTTP状态码比例，从而知道nginx服务状况
- 统计响应时间，判断超时请求
 - 根据经验，如果响应时间分布集中在某个数字，并且标准差很小，则很可能是因为超时。
 - 比如5.032, 5.045, 5.012 ... 则很可能是因为5s超时
- 统计QPS， 对比负载是否均衡

 基础架构快报

access_log 排查

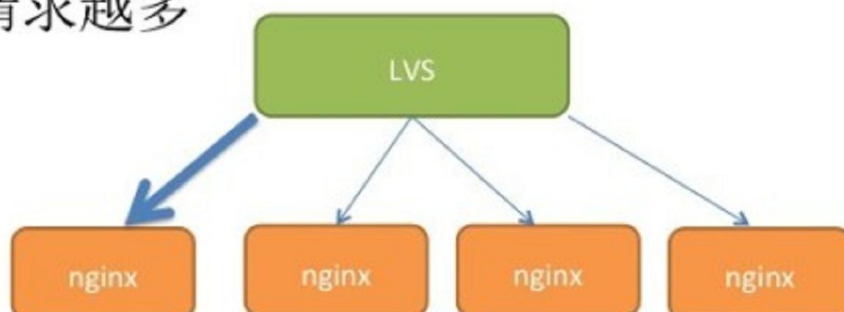
在一次nginx灰度过程中，我们选取了一台nginx上线，在接下来的半小时内，我们发现nginx占用内存不断增大，直到耗尽系统资源。

- 排查
 - Error_log 中没有发现错误信息
 - 监控发现新上线nginx流量大增
 - 分析access_log, 发现新上线nginx的QPS是原机器的6-8倍
- Keepalive 导致流量负载不均衡
 - LVS 作为负载均衡server时，后端的nginx设置的keepalive_timeout时间不能差异过大

基础架构快报

LVS负载均衡示意图

- LVS是TCP层的负载均衡，只能做连接的均衡
- 连接保持时间越长，导致路由到这个连接请求越多



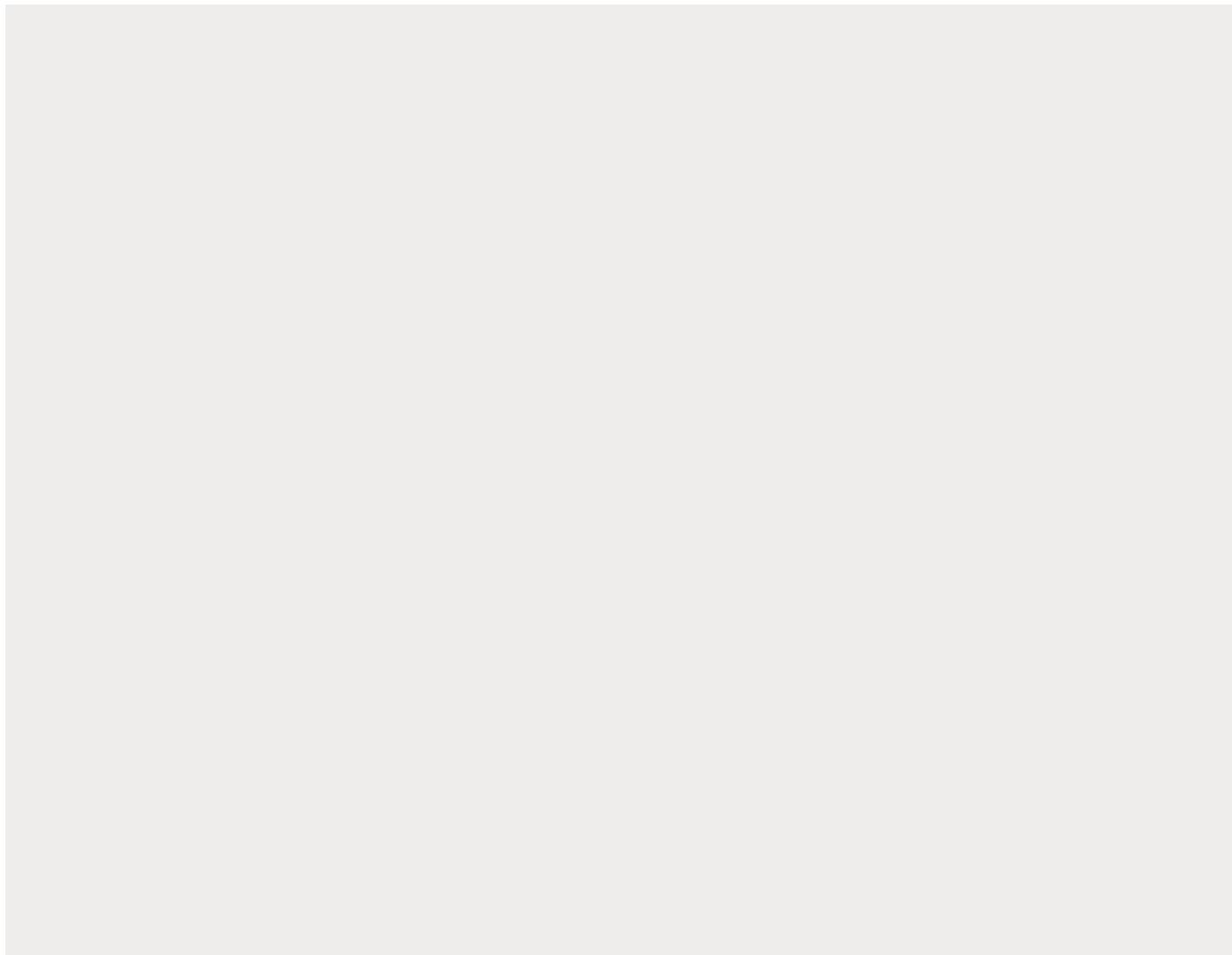
基础架构快报

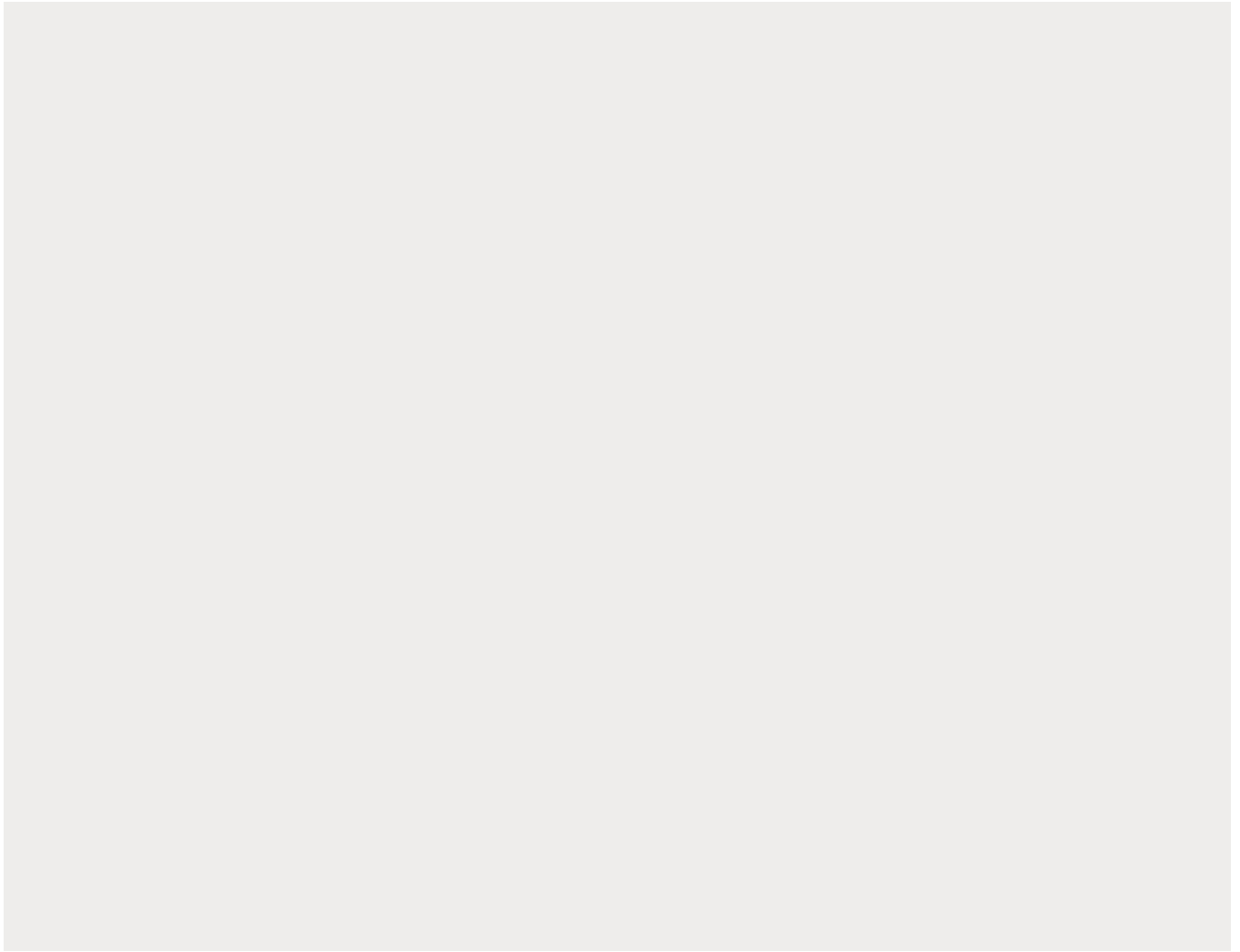
access_log 排查

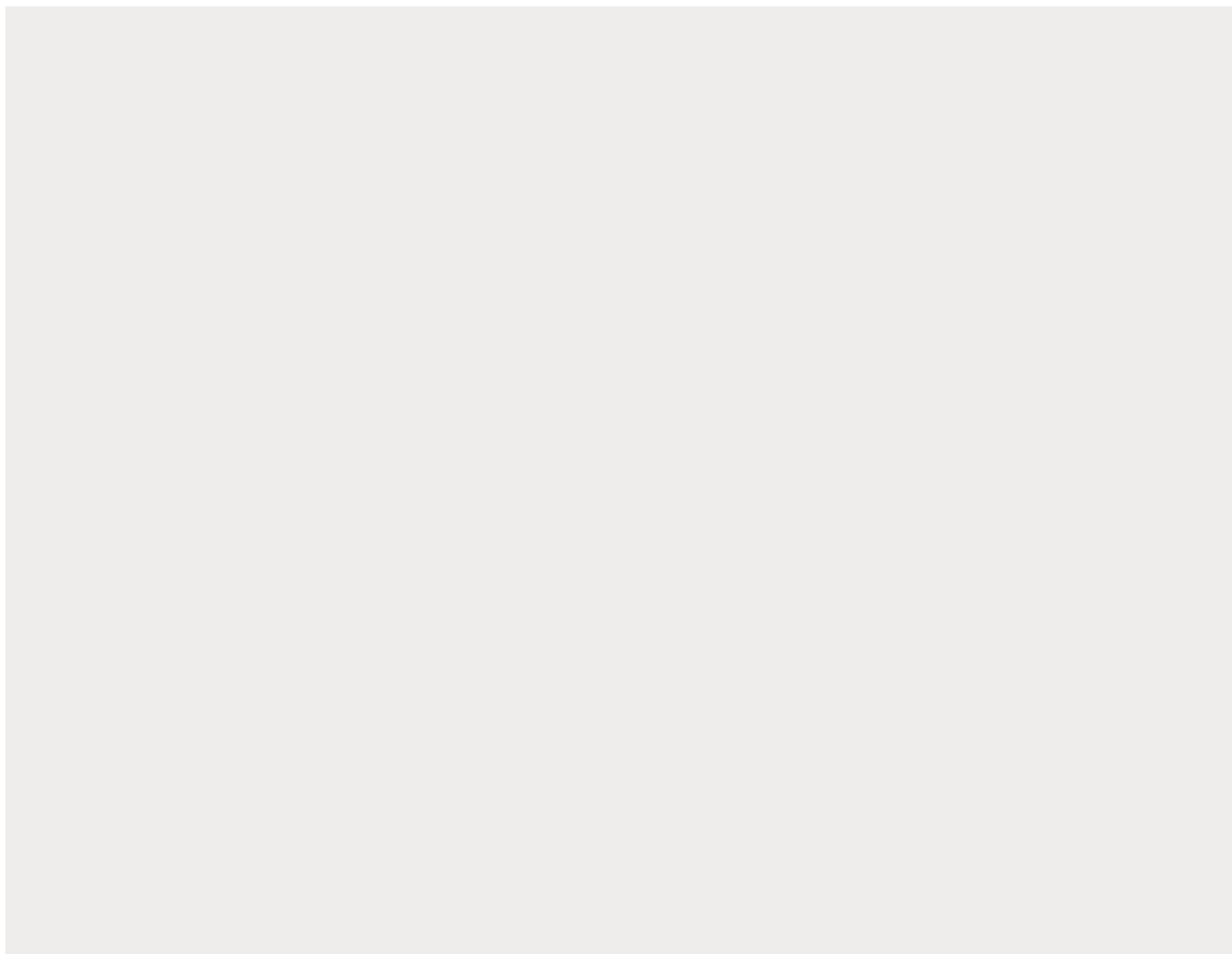
Nginx做反向代理，有时发现某些接口较慢，通常在3s左右

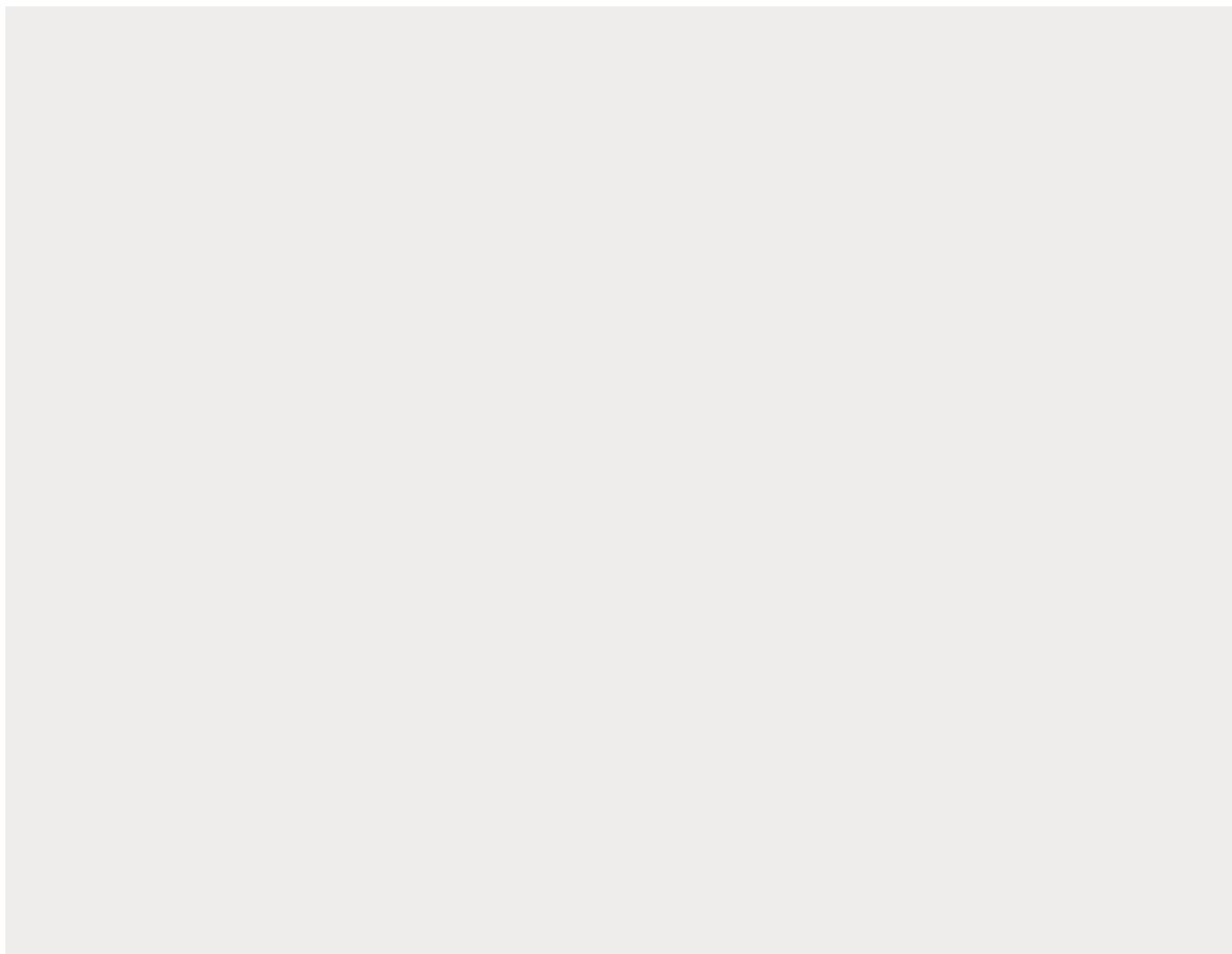
- Nginx access_log 可以记录 upstream_response_time
 - 分析access_log 的upstream_response_time，发现响应大约3s多
 - 分析后端access_log，处理时间在毫秒级别
 - 最终发现，问题是由于nginx跟后端connect较慢导致

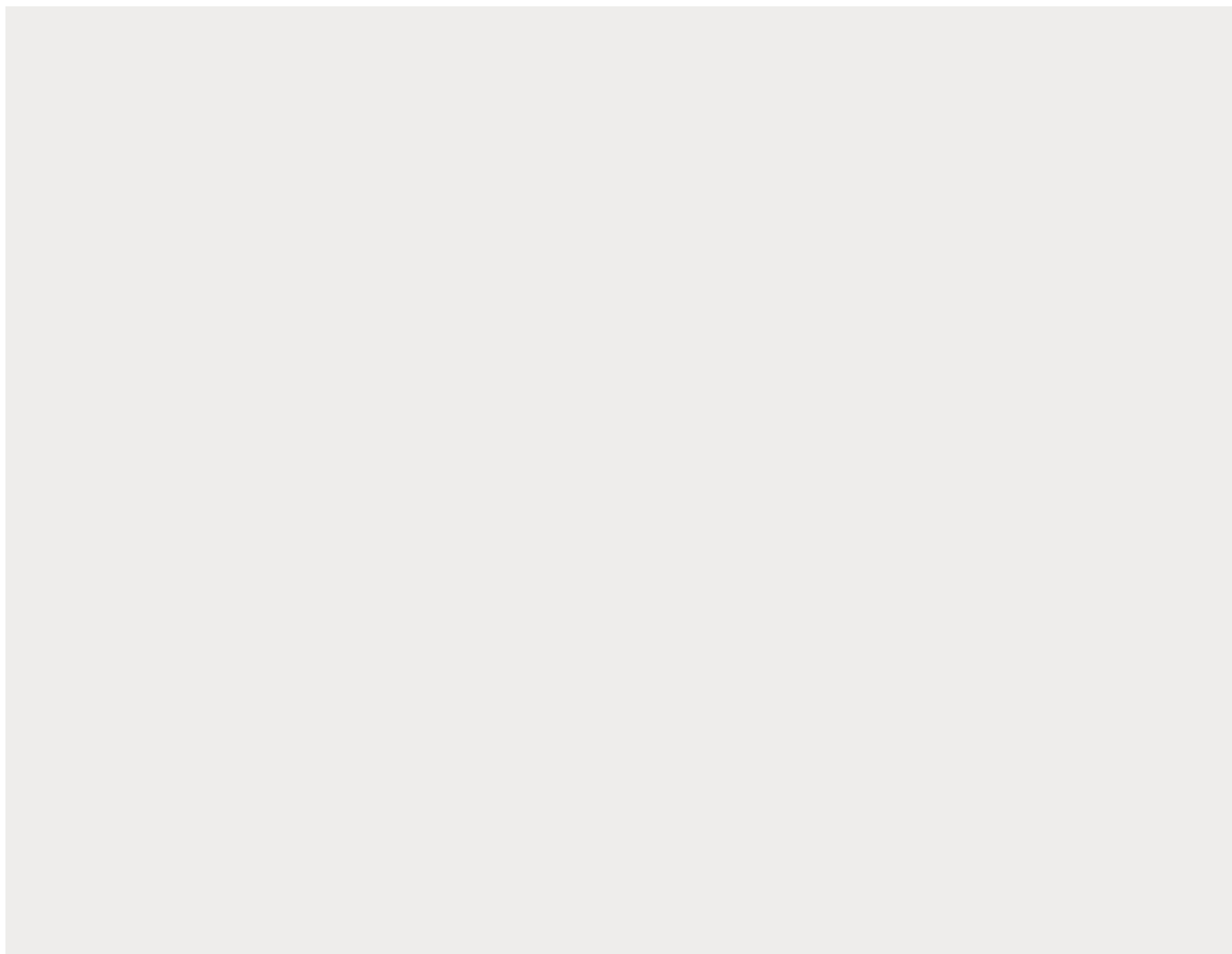
 基础架构快报

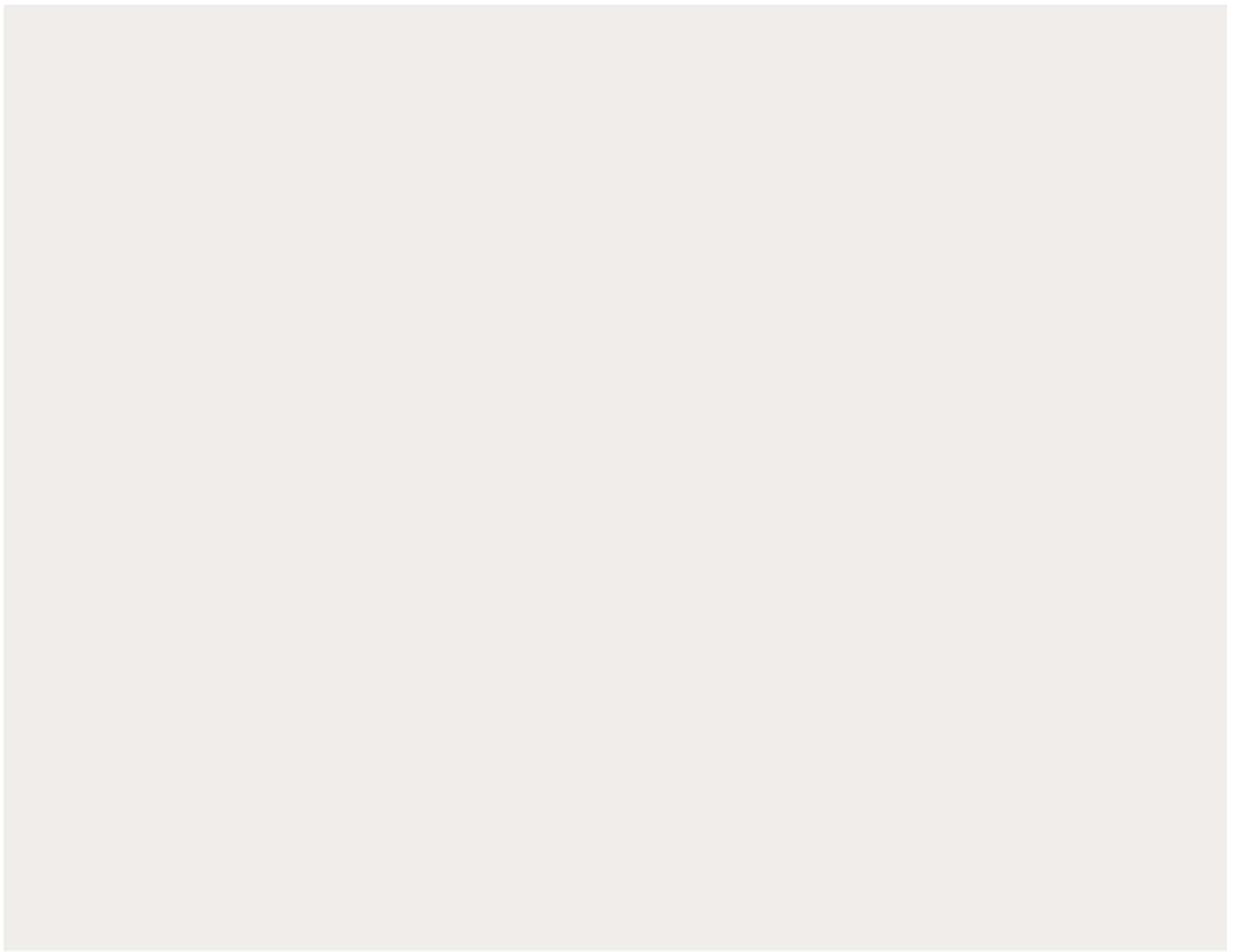


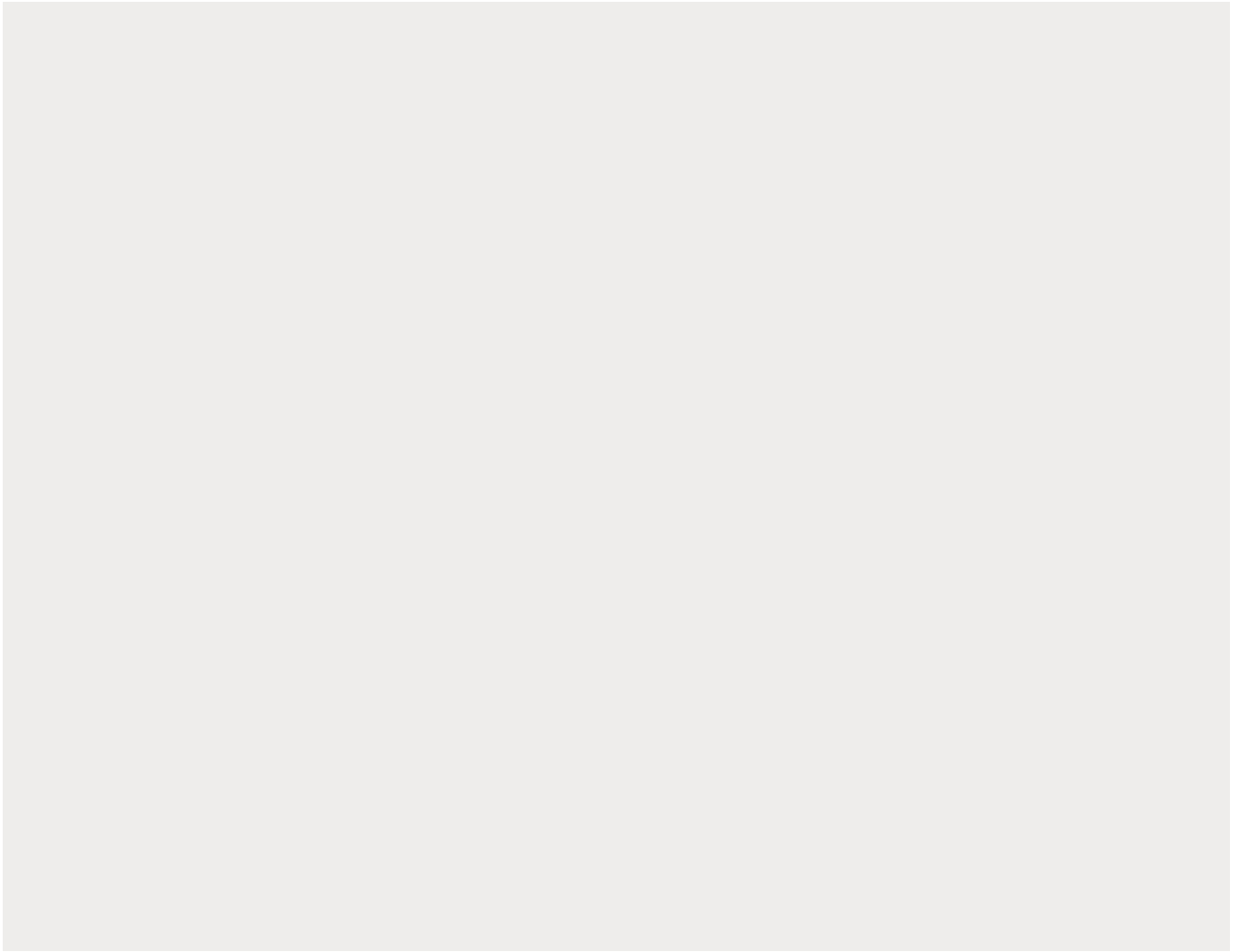


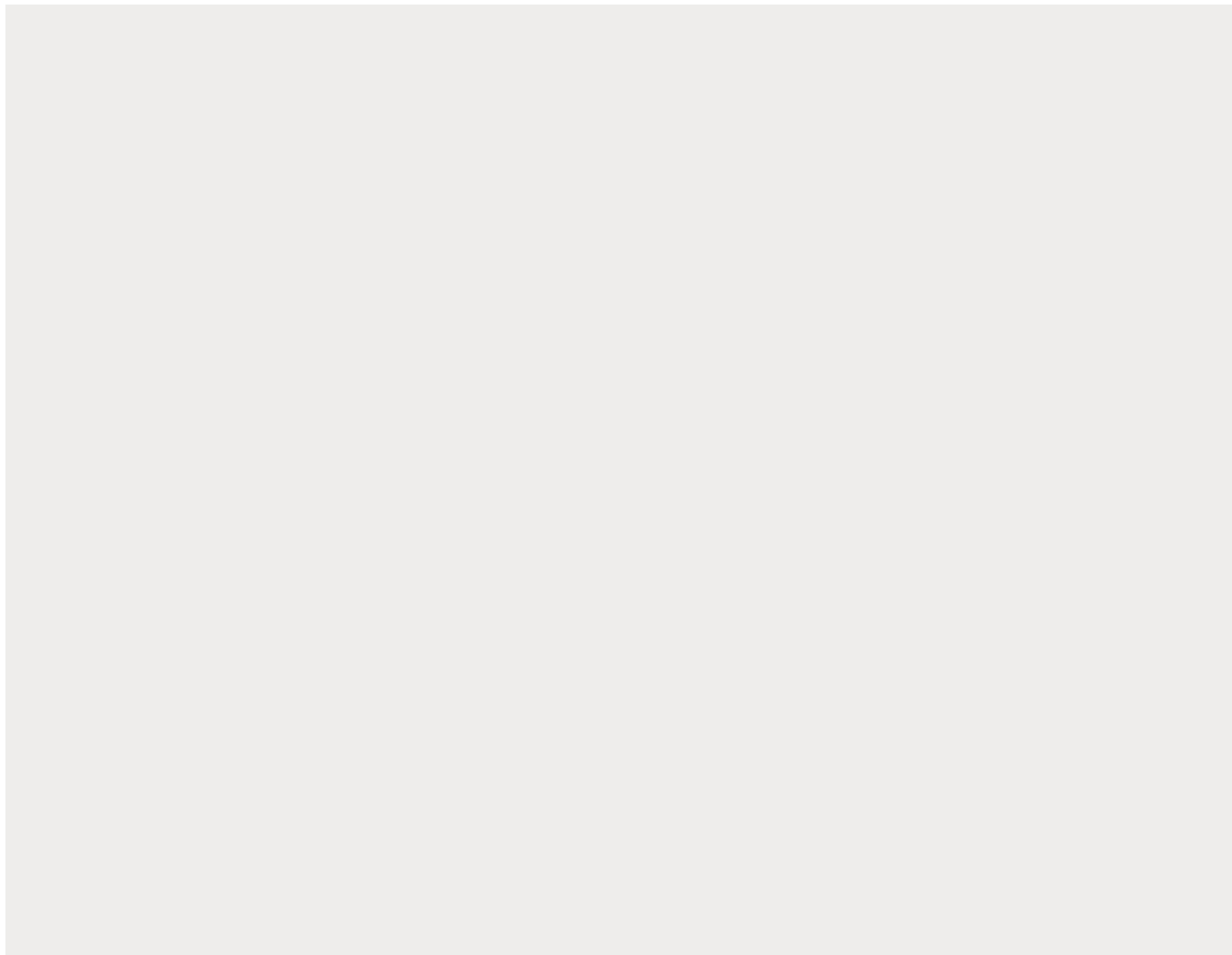


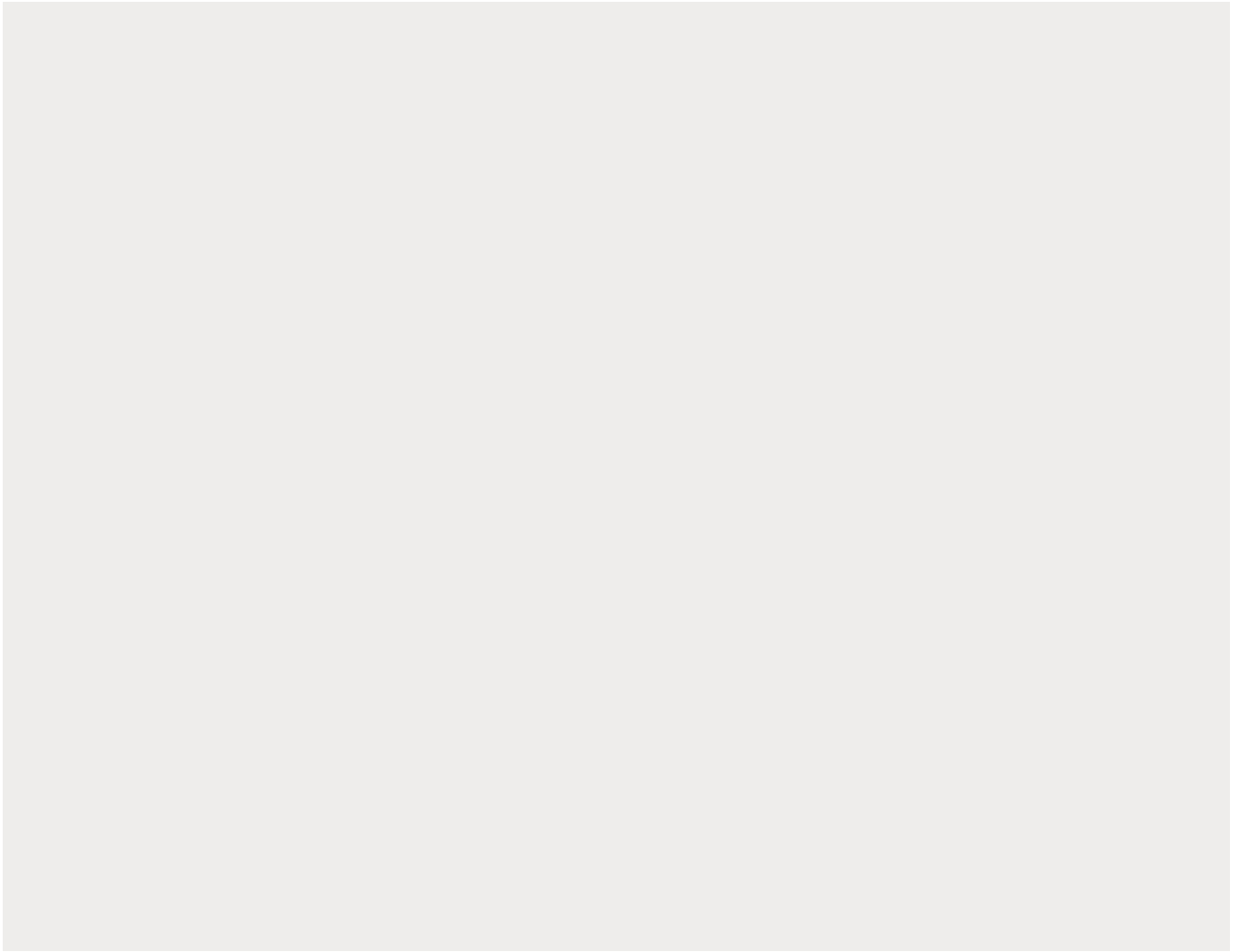


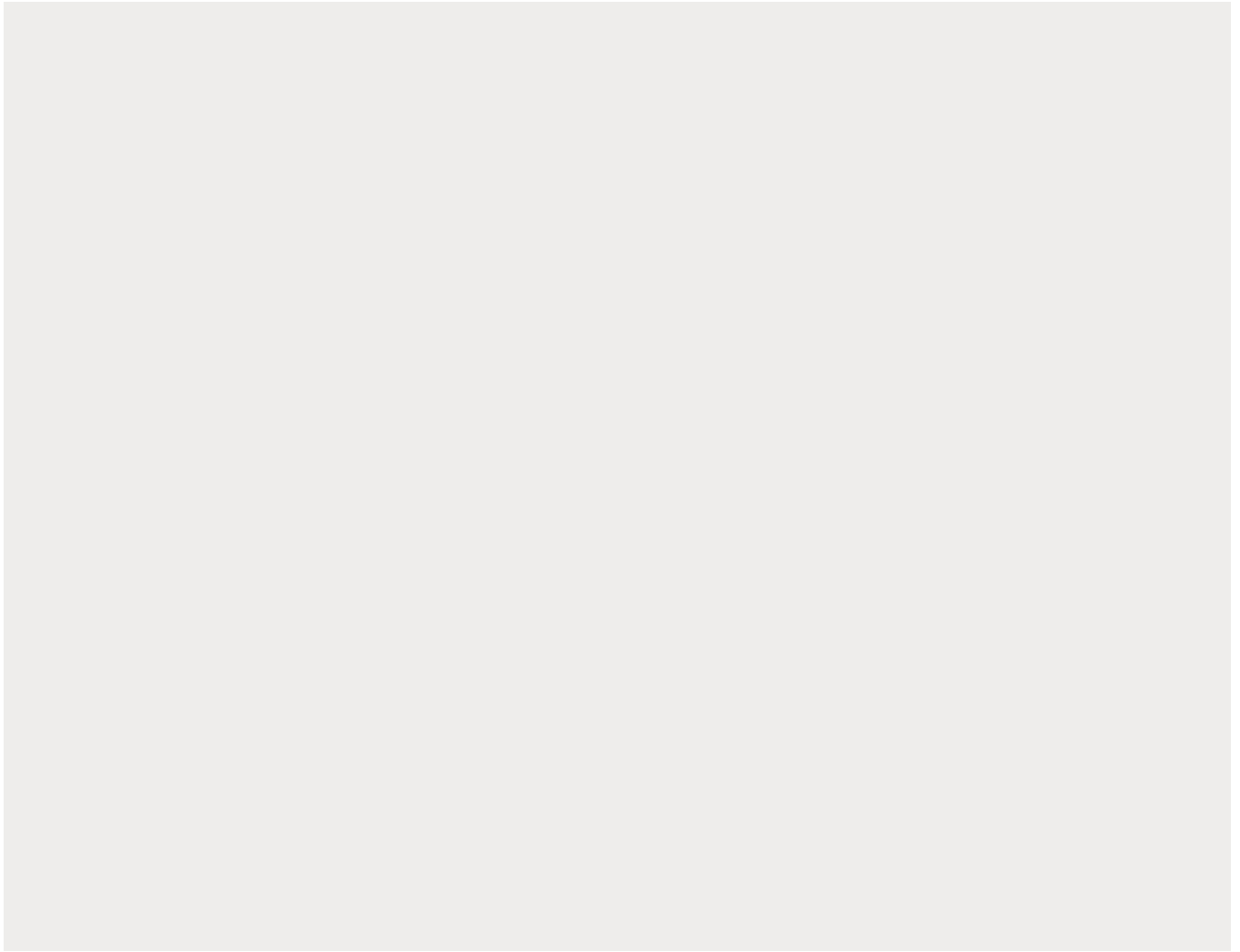


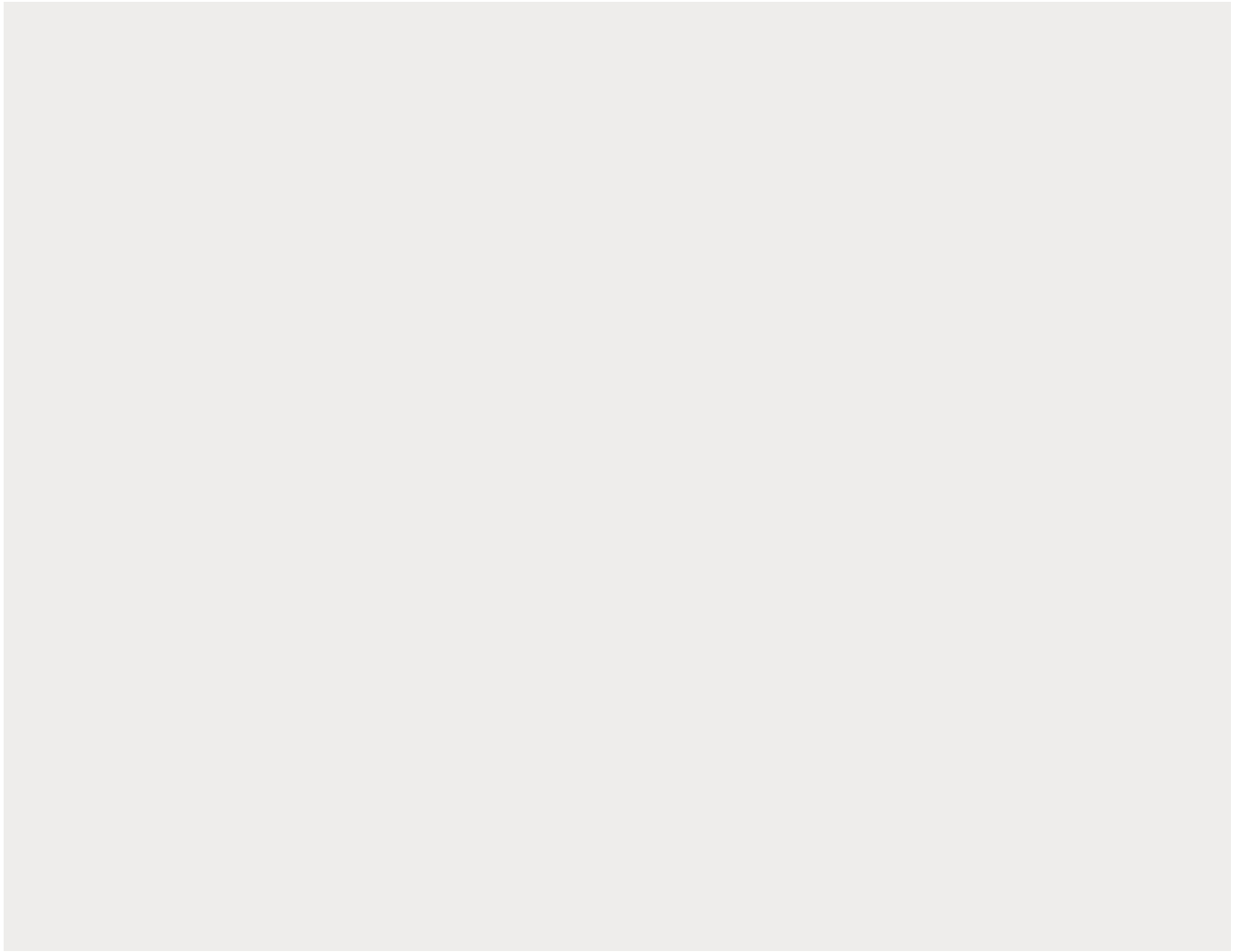


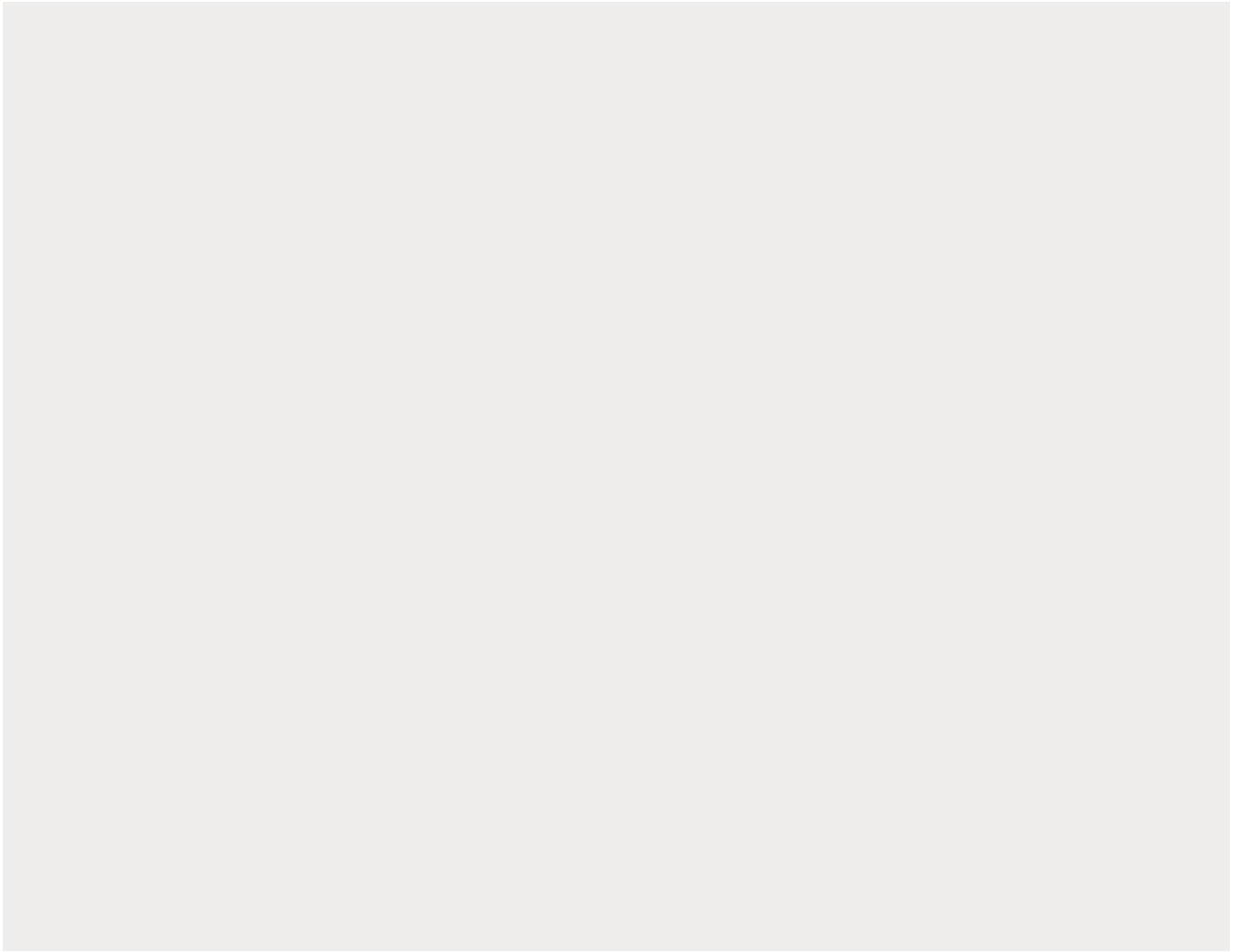


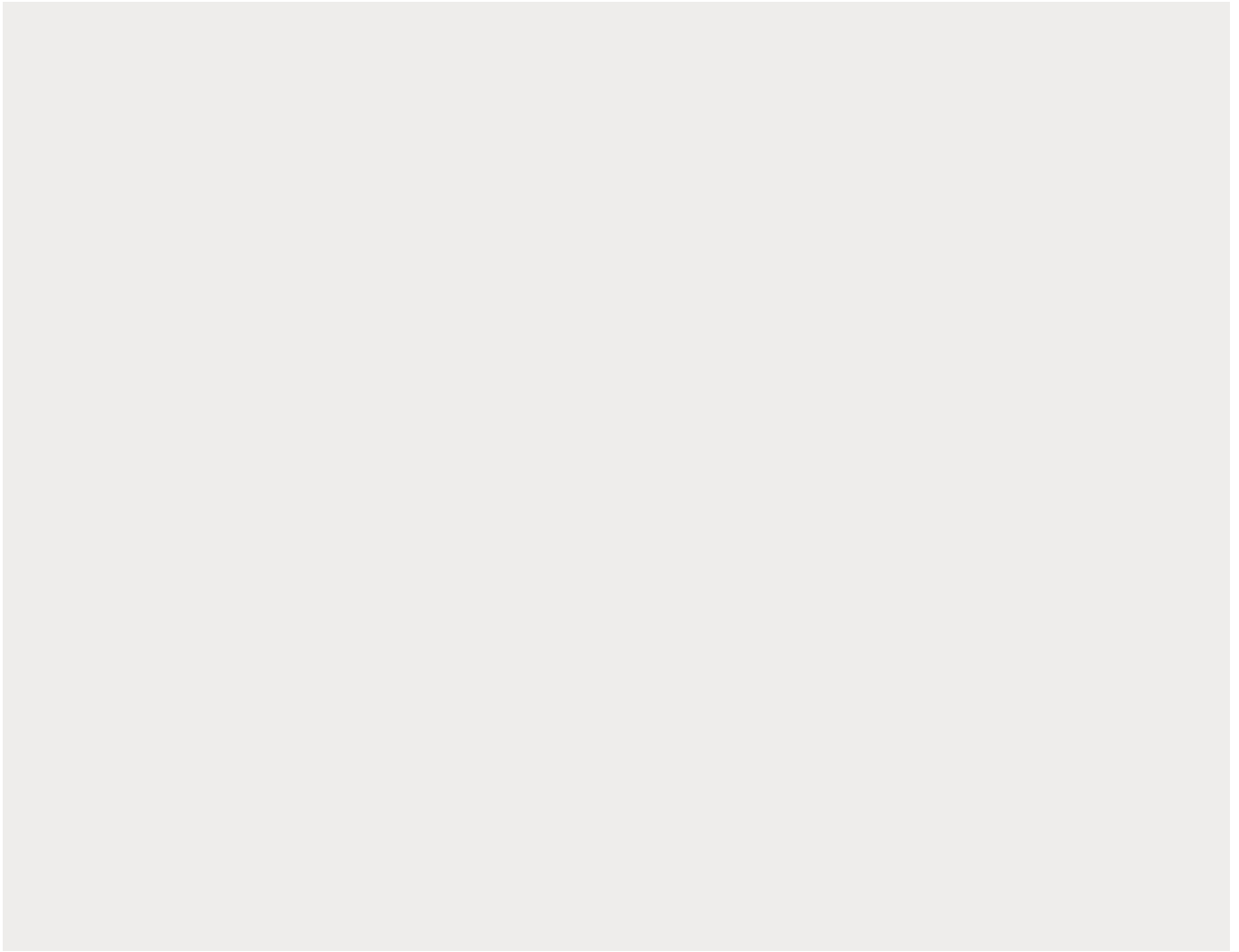


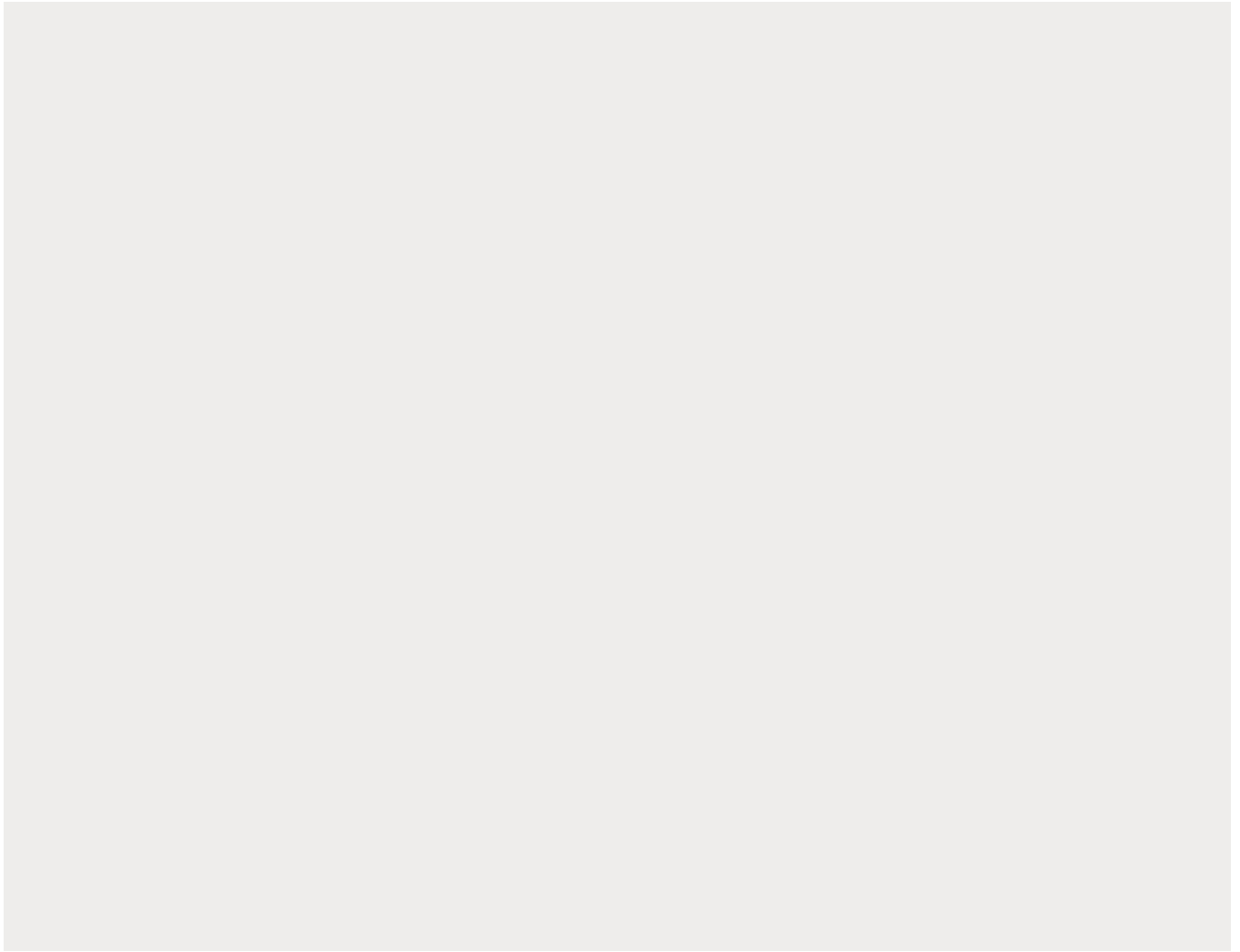


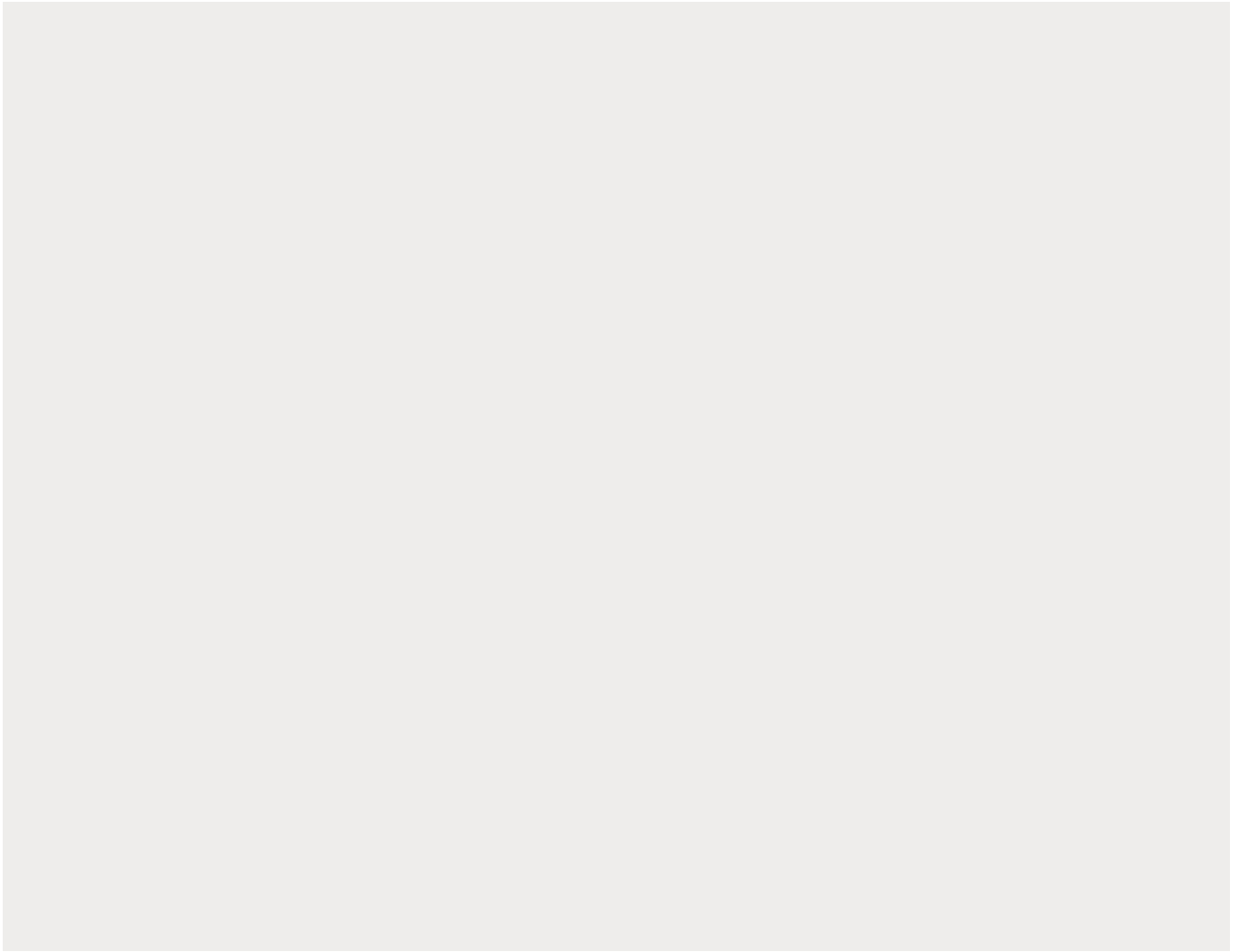


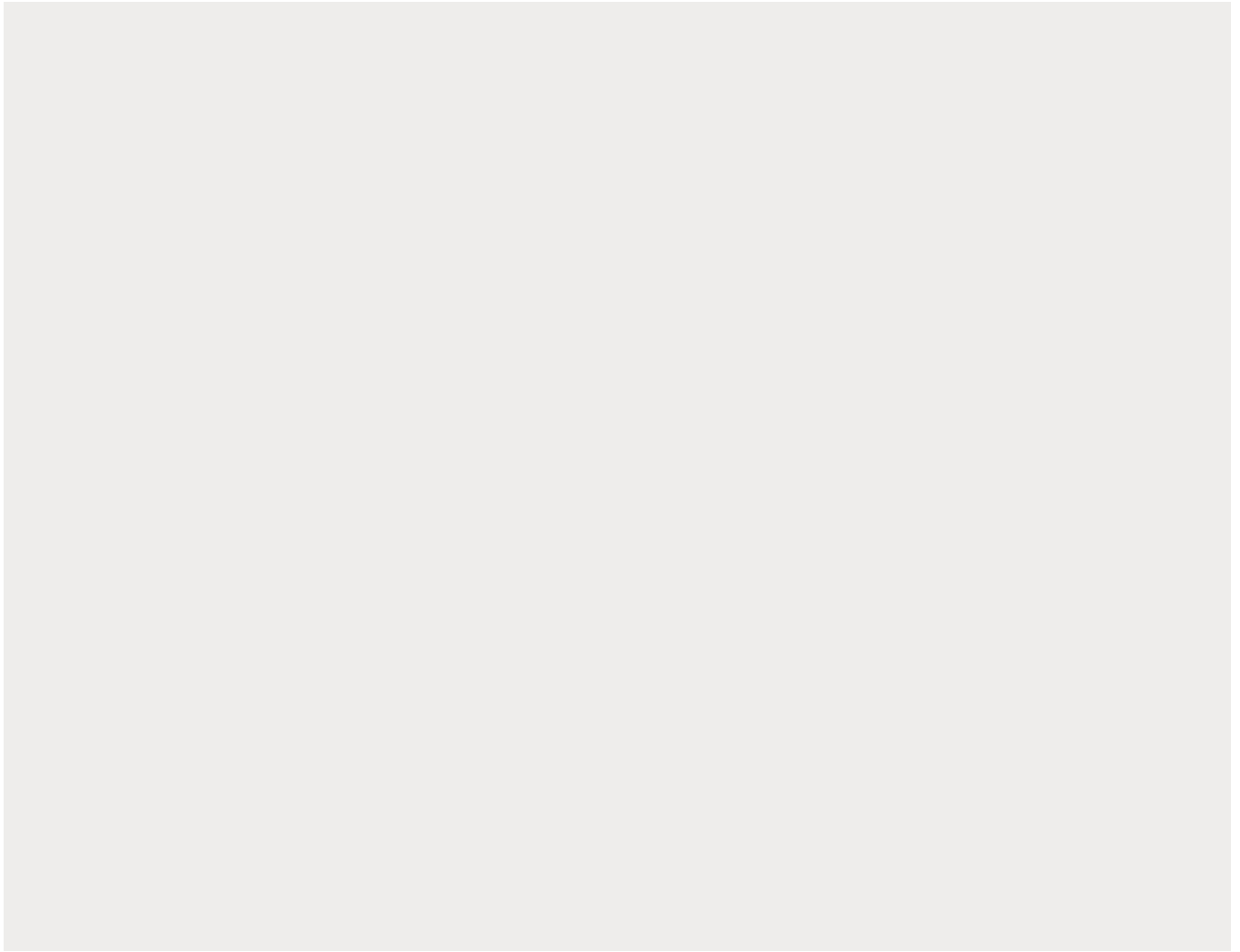












=====

提前祝大家节日快乐！吃好，玩好，线上无事故！（这几天就别上线了...）

另外，基础架构团队为了坚持**节假日不上线**的原则，本篇文章将是十一回归前的最后一篇，咱们两周后再见！

Report